

CFM 1206
U.S. EP

A.N.S. 3/225,163
GAL 2/62

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出願年月日
Date of Application: 1997年 5月16日

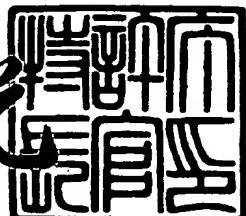
出願番号
Application Number: 平成 9年特許願第127652号

出願人
Applicant(s): キヤノン株式会社

1998年 3月13日

特許庁長官
Commissioner,
Patent Office

荒井寿光



出証番号 出証特平10-3014073

【書類名】 特許願
【整理番号】 3511025
【提出日】 平成 9年 5月16日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 9/00
【発明の名称】 データ転送装置、データ転送システムおよびその方法、
画像処理装置、並びに、記録媒体
【請求項の数】 25
【発明者】
【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会
社内
【氏名】 福長 耕司
【発明者】
【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会
社内
【氏名】 小林 真琴
【発明者】
【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会
社内
【氏名】 片野 清
【発明者】
【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会
社内
【氏名】 鈴木 尚久
【発明者】
【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会
社内
【氏名】 立山 二郎

【発明者】

【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

【氏名】 中村 敦

【特許出願人】

【識別番号】 000001007

【氏名又は名称】 キヤノン株式会社

【代理人】

【識別番号】 100076428

【弁理士】

【氏名又は名称】 大塚 康徳

【電話番号】 03-5276-3241

【選任した代理人】

【識別番号】 100093908

【弁理士】

【氏名又は名称】 松本 研一

【電話番号】 03-5276-3241

【選任した代理人】

【識別番号】 100101306

【弁理士】

【氏名又は名称】 丸山 幸雄

【電話番号】 03-5276-3241

【手数料の表示】

【予納台帳番号】 003458

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

特平 9-127652

【包括委任状番号】 9704672

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ転送装置、データ転送システムおよびその方法、画像処理装置、並びに、記録媒体

【特許請求の範囲】

【請求項1】 シリアルバスにより接続されるホストデバイスとターゲットデバイスとの間で利用されるデータ転送方法であって、

同期転送または非同期転送により前記ホストデバイスから前記ターゲットデバイスへ送信すべきデータを転送するとともに、前記データの転送のための手順信号を共通の非同期転送によって転送することを特徴とするデータ転送方法。

【請求項2】 前記非同期転送には、プッシュバッファモデルおよびプルバッファモデルが含まれることを特徴とする請求項1に記載されたデータ転送方法。

【請求項3】 前記ホストデバイスは、前記共通の非同期転送に基づき前記ターゲットデバイスのためのデータ転送方法を設定することを特徴とする請求項1または請求項2に記載されたデータ転送方法。

【請求項4】 前記ホストデバイスは、前記共通の非同期転送に基づき前記同期転送または非同期転送を選択することを特徴とする請求項1から請求項3の何れかに記載されたデータ転送方法。

【請求項5】 前記シリアルバスはIEEE1394規格に適合または準拠するバスであることを特徴とする請求項1から請求項4の何れかに記載されたデータ転送方法。

【請求項6】 前記シリアルバスはUSB規格に適合または準拠するバスであることを特徴とする請求項1から請求項4の何れかに記載されたデータ転送方法。

【請求項7】 前記ホストデバイスは画像データを出力するデバイスであることを特徴とする請求項1から請求項6の何れかに記載されたデータ転送方法。

【請求項8】 前記ターゲットデバイスは、記録媒体上に画像データに基づく可視像を形成するデバイスであることを特徴とする請求項1から請求項7の何れかに記載されたデータ転送方法。

【請求項9】 請求項1から請求項6の何れかに記載されたデータ転送方法により、前記ターゲットデバイスへ画像データを送信することを特徴とする画像処理装置。

【請求項10】 請求項1から請求項6の何れかに記載されたデータ転送方法により、前記ホストデバイスから受信した画像データを処理することを特徴とする画像処理装置。

【請求項11】 シリアルバスに接続されるデータ転送装置であって、
共通の非同期転送によりデータ転送のための手順信号を転送する転送手段と、
同期転送または非同期転送により、ターゲットデバイスへ送信すべきデータを
送信する送信手段とを有することを特徴とするデータ転送装置。

【請求項12】 前記同期転送には、プッシュバッファモデルおよびプルバッファモデルが含まれることを特徴とする請求項11に記載されたデータ転送装置。

【請求項13】 前記送信手段は、前記共通の非同期転送に基づき前記ターゲットデバイスのためのデータ転送方法を設定することを特徴とする請求項11または請求項12に記載されたデータ転送装置。

【請求項14】 前記送信手段は、前記共通の非同期転送に基づき前記同期転送または非同期転送を選択することを特徴とする請求項11から請求項13の何れかに記載されたデータ転送装置。

【請求項15】 前記データ転送されるデータは画像データであることを特徴とする請求項11から請求項14の何れかに記載されたデータ転送装置。

【請求項16】 シリアルバスに接続されるデータ転送装置であって、
共通の非同期転送によりデータ転送のための手順信号を転送する転送手段と、
同期転送または非同期転送により、ホストデバイスからデータを受信する受信
手段とを有することを特徴とするデータ転送装置。

【請求項17】 前記非同期転送には、プッシュバッファモデルおよびプルバッファモデルが含まれることを特徴とする請求項16に記載されたデータ転送装置。

【請求項18】 前記ホストデバイスは、前記共通の非同期転送に基づき前記受信手段のためのデータ転送方法を設定することを特徴とする請求項16または請求項17に記載されたデータ転送装置。

【請求項19】 前記ホストデバイスは、前記共通の非同期転送に基づき前記同期転送または非同期転送を選択することを特徴とする請求項16から請求項18の何れかに記載されたデータ転送装置。

【請求項 20】 さらに、前記転送手段により受信されたデータに基づく可視像を記録媒体上に形成する形成手段を有することを特徴とする請求項16から請求項19の何れかに記載されたデータ転送装置。

【請求項 21】 シリアルバスを介してデータを転送するデータ転送システムであって、

共通の非同期転送によりデータ転送のための手順信号を転送させる第一の転送手段と、

同期転送または非同期転送により、ホストデバイスとターゲットデバイスとの間でデータ転送を行わせる第二の転送手段とを有することを特徴とするデータ転送システム。

【請求項 22】 前記非同期転送には、プッシュバッファモデルおよびプルバッファモデルが含まれることを特徴とする請求項21に記載されたデータ転送システム。

【請求項 23】 前記ホストデバイスは、前記共通の非同期転送に基づき前記ターゲットデバイスのためのデータ転送方法を設定することを特徴とする請求項21または請求項22に記載されたデータ転送装置。

【請求項 24】 前記ホストデバイスは、前記共通の非同期転送に基づき前記同期転送または非同期転送を選択することを特徴とする請求項21から請求項23の何れかに記載されたデータ転送装置。

【請求項 25】 シリアルバスにより直結されるホストデバイスとターゲットデバイスとの間で利用されるデータ転送方法のプログラムコードが格納された記録媒体であって、

共通の非同期転送によりデータ転送のための手順信号を転送させるための第一の転送ステップのコードと、

同期転送または非同期転送により、ホストデバイスとターゲットデバイスとの間でデータ転送を行わせるための第二の転送ステップのコードとを有することを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明はデータ転送装置、データ転送システムおよびその方法、画像処理装置、並びに、記録媒体に関し、例えば、IEEE1394などにより規定されるシリアルインターフェイスを介して、ディジタルカメラなどの画像供給デバイスとプリンタなどの画像処理デバイスとを直結する場合のデータ転送装置、データ転送システムおよびその方法、画像処理装置、並びに、記録媒体に関するものである。

【0002】**【従来の技術】**

プリンタは、セントロニクスやRS232Cと言ったパラレルあるいはシリアルインターフェイスを介して、ホストデバイスであるパーソナルコンピュータ(PC)と接続されている。

【0003】

また、スキャナ、ディジタルスチルカメラ、ディジタルビデオカメラといった画像供給デバイスであるディジタル機器もPCに接続されている。各ディジタル機器により取込まれた画像データは、一旦PC上のハードディスクなどに取込まれた後、PC上のアプリケーションソフトウェアなどにより処理されてプリンタ用の印刷データに変換され、上記のインターフェイスを経由してプリンタに送られる。

【0004】

上記のようなシステムでは、各ディジタル機器やプリンタなどを制御するためのドライバソフトウェアがそれぞれ独立にPCに存在し、ディジタル機器から出力された画像データは、それらドライバソフトウェアによりPC上で使い易くかつ表示し易い形式のデータとして保存される。保存されたデータは、入力機器の画像特性と出力機器の画像特性とを考慮した画像処理方法により印刷データに変換される。

【0005】

今日、IEEE1394により規定されるインターフェイス（以下「1394シリアルバス」と呼ぶ）のような新しいインターフェイスでは、画像供給デバイスとプリンタとを直結することも可能である。1394シリアルバスにより画像供給デバイスとプリンタとを直結する場合、FCP(Function Control Protocol)のオペランドに印刷データ

タを含める方法が考えられる。また、1394シリアルバスでは、データ転送のためのレジスタ領域を設けて、そのレジスタ領域にデータを書込むことでデータ転送を行う方法も考えられる。

【0006】

また、1394シリアルバスには、データ転送用の制御手順として複数の方法があり、それぞれの機器に適した方式によるデータ転送を行うことができる。

【0007】

また、1394シリアルバスには、同期転送(isochronous)と非同期転送(asynchronous)の二つの転送モードがあり、同期転送により時間に厳密なデータ、例えばリアルタイムデータの転送を、非同期転送により簡易なデータ転送を実現することができる。

【0008】

【発明が解決しようとする課題】

しかし、上述した技術においては、次のような問題点がある。前述したように、画像供給デバイスから出力される画像データは、PCにより印刷データに変換されてプリンタにより印刷されるものであるから、画像供給デバイスとプリンタを直結したとしても、PCが無ければ印刷を行うことができない。ディジタルビデオカメラから出力される画像データを直接印刷するビデオプリンタと呼ばれるプリンタもあるが、特定の機種間で接続ができるだけであり、多数の画像供給デバイスと直結して使える汎用性の高いビデオプリンタはない。つまり、1394シリアルバスなどの特徴であるデバイス間を直結する機能を生かし、画像供給デバイスからプリンタへ画像データを直接送って印刷することはできない。

【0009】

1394シリアルバスにより画像供給デバイスとプリンタを直結し、FCPのオペランドに印刷データを含める前述した方法は、制御コマンドと印刷データとを分離することができない問題がある上、コマンドに対して常にレスポンスが必要なため転送効率が低いという問題もある。また、前述したデータ転送のためのレジスタ領域を設ける方法は、そのレジスタ領域にデータを書込むことが可能であるかどうかを判定するための処理がデータ転送の度に必要になる。従って、この判定

処理のオーバヘッドが大きくなり、やはり転送効率が低下するという問題が発生する。

【0010】

また、同期転送モードと非同期転送モードでは基本的な転送方法が異なるため、同じ転送手順で同期転送モードと非同期転送モードを切換えることは難しい。

【0011】

本発明は、上述した問題を個々にまたはまとめて解決するためのものであり、1394シリアルバスなどによりホストデバイスとターゲットデバイスを直結し、ホストデバイスから直接ターゲットデバイスへ送られる画像データの処理をターゲットデバイスに行わせるのに適したデータ転送装置、データ転送システムおよびその方法、画像処理装置、並びに、記録媒体を提供することを目的とする。

【0012】

また、制御コマンドとデータの分離が可能で転送効率のよいデータ転送装置、データ転送システムおよびその方法、画像処理装置、並びに、記録媒体を提供することを他の目的とする。

【0013】

また、同期転送と非同期転送と同じ転送手順で行うことができるデータ転送装置、データ転送システムおよびその方法、画像処理装置、並びに、記録媒体を提供することを他の目的とする。

【0014】

【課題を解決するための手段】本発明は、前記の目的を達成する一手段として、以下の構成を備える。

本発明にかかるデータ転送装置は、シリアルバスに接続されるデータ転送装置であって、共通の非同期転送によりデータ転送のための手順信号を転送する転送手段と、同期転送または非同期転送により、ターゲットデバイスへ送信すべきデータを送信する送信手段とを有することを特徴とする。

【0015】

また、シリアルバスに接続されるデータ転送装置であって、共通の非同期転送によりデータ転送のための手順信号を転送する転送手段と、同期転送または非同

期転送により、ホストデバイスからデータを受信する受信手段とを有することを特徴とする。

【0016】

本発明にかかるデータ転送方法は、シリアルバスにより接続されるホストデバイスとターゲットデバイスとの間で利用されるデータ転送方法であって、同期転送または非同期転送により前記ホストデバイスから前記ターゲットデバイスへ送信すべきデータを転送するとともに、前記データの転送のための手順信号を共通の非同期転送によって転送することを特徴とする。

【0017】

本発明にかかるデータ転送システムは、シリアルバスを介してデータを転送するデータ転送システムであって、共通の非同期転送によりデータ転送のための手順信号を転送させる第一の転送手段と、同期転送または非同期転送により、ホストデバイスとターゲットデバイスとの間でデータ転送を行わせる第二の転送手段とを有することを特徴とする。

【0018】

本発明にかかる画像処理装置は、シリアルバスにより接続されるホストデバイスとターゲットデバイスとの間で利用されるデータ転送方法であって、同期転送または非同期転送により前記ホストデバイスから前記ターゲットデバイスへ送信すべきデータを転送するとともに、前記データの転送のための手順信号を共通の非同期転送によって転送するデータ転送方法により、前記ターゲットデバイスへ画像データを送信することを特徴とする。

【0019】

また、シリアルバスにより接続されるホストデバイスとターゲットデバイスとの間で利用されるデータ転送方法であって、同期転送または非同期転送により前記ホストデバイスから前記ターゲットデバイスへ送信すべきデータを転送するとともに、前記データの転送のための手順信号を共通の非同期転送によって転送するデータ転送方法により、前記ホストデバイスから受信した画像データを処理することを特徴とする。

【0020】

【発明の実施の形態】

以下、本発明にかかる一実施形態のデータ転送方法を図面を参照して詳細に説明する。

【0021】

図1は本発明を適用するシステムの一般的な構成例を示す図で、PC103、プリンタ102およびディジタルビデオカメラ(DVC)101を1394シリアルバスを用いて接続するものである。そこで、予め、1394シリアルバスの概要を説明をする。

【0022】

【IEEE1394の概要】

家庭用ディジタルVTRやディジタルビデオディスク(DVD)の登場に伴い、ビデオデータやオーディオデータ（以下、まとめて「AVデータ」と呼ぶ）など、リアルタイムかつ情報量の多いデータを転送する必要が生じている。AVデータをリアルタイムに、PCへ転送したり、その他のディジタル機器に転送するには、高速のデータ転送能力をもつインターフェイスが必要になる。そういった観点から開発されたインターフェイスが1394シリアルバスである。

【0023】

図2に1394シリアルバスを用いて構成されるネットワークシステムの例を示す。このシステムは機器AからHを備え、A-B間、A-C間、B-D間、D-E間、C-F間、C-G間、およびC-H間がそれぞれ1394シリアルバス用のツイストペアケーブルで接続されている。これらの機器AからHの例としては、パソコンなどのホストコンピュータ装置、および、コンピュータ周辺機器である。コンピュータ周辺機器としては、ディジタルVCR、DVDプレーヤ、ディジタルスチルカメラ、ハードディスクや光ディスクなどのメディアを用いる記憶装置、CRTやLCDのモニタ、チューナ、イメージスキャナ、フィルムスキャナ、プリンタ、MODEM、ターミナルアダプタ(TA)などコンピュータ周辺機器のすべてが対象になる。なお、プリンタの記録方式は、レーザビームやLEDを用いた電子写真方式、インクジェット方式、インク溶融型や昇華型の熱転写方式、感熱記録方式など、どんな方式でも構わない。

【0024】

各機器間の接続は、ディジーチェーン方式とノード分岐方式との混在が可能で

あり、自由度の高い接続を行うことができる。また、各機器はそれぞれIDを有し、互いにIDを認識し合うことによって、1394シリアルバスで接続された範囲において、一つのネットワークを構成している。例えば、各機器間をそれぞれ一本の1394シリアルバス用ケーブルでディジーチェーン接続するだけで、それぞれの機器が中継の役割を担うので、全体として一つのネットワークを構成することができる。

【0025】

また、1394シリアルバスはPlug and Play機能に対応し、1394シリアルバス用ケーブルを機器に接続するだけで自動的に機器を認識し、接続状況を認識する機能を有している。また、図2に示すようなシステムにおいて、ネットワークからある機器が外されたり、または新たに加えられたときなど、自動的にバスをリセット（それまでのネットワークの構成情報をリセット）して、新たなネットワークを再構築する。この機能によって、その時々のネットワークの構成を常時設定、認識することができる。

【0026】

また、1394シリアルバスのデータ転送速度は、100/200/400Mbpsが定義されていて、上位の転送速度をもつ機器が下位の転送速度をサポートすることで、互換性が保たれている。データ転送モードとしては、コントロール信号などの非同期データを転送する非同期(Asynchronous)転送モード(ATM)と、リアルタイムなAVデータ等の同期データを転送する同期(Isochronous)転送モードがある。この非同期データと同期データは、各サイクル（通常 $125\mu s$ /サイクル）の中で、サイクル開始を示すサイクルスタートパケット(CSP)の転送に続き、同期データの転送を優先しつつ、サイクル内で混在して転送される。

【0027】

図3は1394シリアルバスの構成例を示す図である。1394シリアルバスはレイヤ構造で構成されている。図3に示すように、コネクタポート810には、1394シリアルバス用のケーブル813の先端のコネクタが接続される。コネクタポート810の上位には、ハードウェア部800で構成されるフィジカルレイヤ811とリンクレイヤ812がある。ハードウェア部800はインタフェイス用チップで構成され、そのうちフ

イジカルレイヤ811は符号化やコネクション関連の制御等を行い、リンクレイヤ812はパケット転送やサイクルタイムの制御等を行う。

【0028】

ファームウェア部801のトランザクションレイヤ814は、転送（トランザクション）すべきデータの管理を行い、Read、Write、Lockの命令を出す。ファームウェア部801のマネージメントレイヤ815は、1394シリアルバスに接続されている各機器の接続状況やIDの管理を行い、ネットワークの構成を管理する。上記のハードウェアとファームウェアまでが、1394シリアルバスの実質的な構成である。

【0029】

また、ソフトウェア部802のアプリケーションレイヤ816は、利用されるソフトによって異なり、インターフェイス上でどのようにしてデータを転送するかは、プリンタやAV/Cプロトコルなどのプロトコルによって定義される。

【0030】

図4は1394シリアルバスにおけるアドレス空間の一例を示す図である。1394シリアルバスに接続された各機器（ノード）には必ずノードに固有の64ビットアドレスをもたせる。そして、このアドレスは機器のメモリに格納されていて、自分や相手のノードアドレスを常時認識することで、通信相手を指定したデータ通信を行うことができる。

【0031】

1394シリアルバスのアドレッシングは、IEEE1212規格に準じた方式であり、アドレス設定は、最初の10ビットがバスの番号の指定用に、次の6ビットがノードIDの指定用に使われる。

【0032】

それぞれの機器内で使用される48ビットのアドレスについても、20ビットと28ビットに分けられ、256Mバイト単位の構造をもって利用される。最初の20ビットのアドレス空間のうち0～0xFFFFDはメモリ空間、0xFFFFEはプライベート空間、0xFFFFFはレジスタ空間とそれぞれ呼ばれる。プライベート空間は機器内で自由に利用できるアドレスであり、レジスタ空間にはバスに接続された機器間で共通な情報が置かれ、各機器間のコミュニケーションに使われる。

【0033】

レジスタ空間の、最初の512バイトにはCSRアーキテクチャのコアになるレジスタ（CSRコア）が、次の512バイトにはシリアルバスのレジスタが、その次の1024バイトにはコンフィグレーションROMが、残りはユニット空間で機器固有のレジスタが、それぞれ置かれる。

【0034】

一般的には異種バスシステムの設計の簡略化のため、ノードは初期ユニット空間の最初の2048バイトだけを使うべきであり、この結果としてCSRコア、シリアルバスのレジスタ、コンフィグレーションROMおよびユニット空間の最初の2048バイトを合わせて4096バイトで構成することが望ましい。

【0035】

以上が、1394シリアルバスの概要である。次に、1394シリアルバスの特徴をより詳細に説明する。

【0036】**【1394シリアルバスの詳細】****【1394シリアルバスの電気的仕様】**

図5は1394シリアルバス用のケーブルの断面を示す図である。1394シリアルバス用ケーブルには、二組のツイストペア信号線の他に、電源ラインが設けられている。これによって、電源を持たない機器や、故障などにより電圧が低下した機器にも電力の供給が可能になる。電源線により供給される直流電力の電圧は8~40V、電流は最大電流1.5Aに規定されている。なお、DVケーブルと呼ばれる規格では、電源ラインを省いた四線で構成される。

【0037】**【DS-Link方式】**

図6は1394シリアルバスで採用されている、データ転送方式のDS-Link(Data/Stripe Link)方式を説明するための図である。

【0038】

DS-Link方式は、高速なシリアルデータ通信に適し、二組の信号線を必要とする。つまり、二組のより対線のうち一組でデータ信号を送り、もう一組でストロー

ーブ信号を送る構成になっている。受信側では、このデータ信号と、ストローブ信号との排他的論理和をとることによってクロックを生成することができるという特徴がある。このため、DS-Link方式を用いると、データ信号中にクロック信号を混入させる必要がないので他のシリアルデータ転送方式に比べて転送効率が高い、クロック信号を生成できるので位相ロックループ(PLL)回路が不要になり、その分コントローラLSIの回路規模を小さくすることができる。さらに、転送すべきデータが無いときにアイドル状態であることを示す情報を送る必要が無いので、各機器のトランシーバ回路をスリープ状態にすることができ、消費電力の低減が図れる、などが挙げられる。

【0039】

[バスリセットのシーケンス]

1394シリアルバスに接続されている各機器（ノード）にはノードIDが与えられ、ネットワークを構成するノードとして認識される。例えば、ネットワーク機器の接続分離や電源のオン/オフなどによるノード数の増減、つまりネットワーク構成に変化があり、新たなネットワーク構成を認識する必要があるとき、その変化を検知した各ノードはバス上にバスリセット信号を送信して、新たなネットワーク構成を認識するモードに入る。このネットワーク構成の変化の検知は、コネクタポート810においてバイアス電圧の変化を検知することによって行われる。

【0040】

あるノードからバスリセット信号が送信されると、各ノードのフィジカルレイヤ811はこのバスリセット信号を受けると同時にリンクレイヤ812にバスリセットの発生を伝達し、かつ他のノードにバスリセット信号を伝達する。最終的にすべてのノードがバスリセット信号を受信した後、バスリセットのシーケンスが起動される。なお、バスリセットのシーケンスは、ケーブルが抜き挿しされた場合や、ネットワークの異常等をハードウェアが検出した場合に起動されるとともに、プロトコルによるホスト制御などフィジカルレイヤ811に直接命令を与えることによっても起動される。また、バスリセットのシーケンスが起動されると、データ転送は、一時中断され、バスリセットの間は待たされ、バスリセット終了後、新しいネットワーク構成の基で再開される。

【0041】

[ノードID決定のシーケンス]

バスリセットの後、各ノードは新しいネットワーク構成を構築するために、各ノードにIDを与える動作に入る。このときの、バスリセットからノードID決定までの一般的なシーケンスを図7から図9に示すフローチャートを用いて説明する。

【0042】

図7は、バスリセット信号の発生から、ノードIDが決定し、データ転送が行えるようになるまでの一連のシーケンス例を示すフローチャートである。各ノードは、ステップS101でバスリセット信号の発生を常時監視し、バスリセット信号が発生するとステップS102に移り、ネットワーク構成がリセットされた状態において新たなネットワーク構成を得るために、互いに直結されているノード間で親子関係が宣言される。そしてステップS103の判定により、すべてのノード間で親子関係が決ったと判定されるまでステップS102が繰り返される。

【0043】

親子関係が決定するとステップS104へ進みルート(root)ノードが決定され、ステップS105で各ノードにIDを与えるノードIDの設定作業が行われる。ルートノードから所定のノード順にノードIDの設定が行われ、ステップS106の判定により、すべてのノードにIDが与えられたと判定されるまでステップS105が繰り返される。

【0044】

ノードIDの設定が終了すると、新しいネットワーク構成がすべてのノードにおいて認識されることになるのでノード間のデータ転送が行える状態になり、ステップS107でデータ転送が開始されるとともに、シーケンスはステップS101へ戻り、再びバスリセット信号の発生が監視される。

【0045】

図8はバスリセット信号の監視(S101)からルートノードの決定(S104)までの詳細例を示すフローチャート、図9はノードID設定(S105,S106)の詳細例を示すフローチャートである。

【0046】

図8において、ステップS201でバスリセット信号の発生が監視され、バスリセット信号が発生すると、ネットワーク構成は一旦リセットされる。次に、ステップS202で、リセットされたネットワーク構成を再認識する作業の第一歩として、各機器はフラグFLをリーフノードであることを示すデータでリセットする。そして、ステップS203で、各機器はポート数、つまり自分に接続されている他ノードの数を調べ、ステップS204で、ステップS203の結果に応じて、これから親子関係の宣言を始めるために、未定義（親子関係が決定されていない）ポートの数を調べる。ここで、未定義ポート数は、バスリセットの直後はポート数に等しいが、親子関係が決定されて行くにしたがって、ステップS204で検知される未定義ポートの数は減少する。

【0047】

バスリセットの直後に親子関係の宣言を行えるのは実際のリーフノードに限られている。リーフノードであるか否かはステップS203のポート数の確認結果から知ることができ、つまりポート数が「1」であればリーフノードである。リーフノードは、ステップS205で、接続相手のノードに対して親子関係の宣言「自分は子、相手は親」を行い動作を終了する。

【0048】

一方、ステップS203でポート数が「2以上」であったノード、つまりブランチノードは、バスリセットの直後は「未定義ポート数>1」であるからステップS206へ進み、フラグFLにブランチノードを示すデータをセットし、ステップS207で他ノードから親子関係が宣言されるのを待つ。他ノードから親子関係が宣言され、それを受けたブランチノードはステップS204に戻り、未定義ポート数を確認するが、もし未定義ポート数が「1」になっていれば残るポートに接続された他ノードに対して、ステップS205で「自分は子、相手は親」の親子関係を宣言することができる。また、未だ未定義ポート数が「2以上」あるブランチノードは、ステップS207で再び他ノードから親子関係が宣言されるのを待つことになる。

【0049】

何れか一つのブランチノード（または例外的に、子宣言を行えるのにもかかわらず、すばやく動作しなかったリーフノード）の未定義ポート数が「0」になる

と、ネットワーク全体の親子関係の宣言が終了したことになり、未定義ポート数が「0」になった唯一のノード、つまりすべてノードの親に決まったノードは、ステップS208でフラグFLにルートノードを示すデータをセットし、ステップS209でルートノードとして認識される。

【0050】

このようにして、バスリセットから、ネットワーク内のすべてのノード間における親子関係の宣言までの手順が終了する。

【0051】

次に、各ノードにIDを与える手順を説明するが、最初にIDの設定を行うことができるのはリーフノードである。そして、リーフ→ブランチ→ルートの順に若い番号（ノード番号：0）からIDを設定する。

【0052】

図9のステップS301で、フラグFLに設定されたデータを基にノードの種類、つまりリーフ、ブランチおよびルートに応じた処理に分岐する。

【0053】

まずリーフノードの場合は、ステップS302でネットワーク内に存在するリーフノードの数（自然数）を変数Nに設定した後、ステップS303で各リーフノードがルートノードに対して、ノード番号を要求する。この要求が複数ある場合、ルートノードはステップS304でアービトリエーションを行い、ステップS305である一つのノードにノード番号を与え、他のノードにはノード番号の取得失敗を示す結果を通知する。

【0054】

ステップS306の判断により、ノード番号を取得できなかったリーフノードは、再びステップS303でノード番号の要求を繰り返す。一方、ノード番号を取得できたりーフノードは、ステップS307で、取得したノード番号を含むID情報をブロードキャストすることで全ノードに通知する。ID情報のブロードキャストが終わるとステップS308で、リーフ数を表す変数Nがデクリメントされる。そして、ステップS309の判定により変数Nが「0」になるまでステップS303からS308の手順が繰り返され、すべてのリーフノードのID情報がブロードキャストされた後、ステッ

PS310へ進んで、ブランチノードのID設定に移る。

【0055】

ブランチノードのID設定もリーフノードとほぼ同様に行われる。まず、ステップS310でネットワーク内に存在するブランチノードの数（自然数）を変数Mに設定した後、ステップS311で各ブランチノードがルートノードに対して、ノード番号を要求する。この要求に対してルートノードは、ステップS312でアビトレーションを行い、ステップS313である一つのブランチノードにリーフノードに続く若い番号を与え、ノード番号を取得できなかったブランチノードには取得失敗を示す結果を通知する。

【0056】

ステップS314の判定により、ノード番号の取得に失敗したことを知ったブランチノードは、再びステップS311でノード番号の要求を繰り返す。一方、ノード番号を取得できたブランチノードはステップS315で、取得したノード番号を含むID情報をブロードキャストすることで全ノードに通知する。ID情報のブロードキャストが終わるとステップS316で、ブランチ数を表す変数Mがデクリメントされる。そして、ステップS317の判定により、変数Mが「0」になるまでステップS311からS316の手順が繰返され、すべてのブランチノードのID情報がブロードキャストされた後、ステップS318へ進んで、ルートノードのID設定に移る。

【0057】

ここまで終了すると、最終的にIDを取得していないノードはルートノードのみなので、ステップS318では、他のノードに与えていない最も若い番号を自分のノード番号に設定し、ステップS319でルートノードのID情報をブロードキャストする。

【0058】

以上で、すべてのノードのIDが設定されるまでの手順が終了する。次に、図10に示すネットワーク例を用いてノードID決定のシーケンスの具体的な手順を説明する。

【0059】

図10に示すネットワークは、ルートであるノードBの下位にはノードAとノード

Cが直結され、ノードCの下位にはノードDが直結され、ノードDの下位にはノードEとノードFが直結された階層構造を有する。この、階層構造やルートノード、ノードIDを決定する手順は以下のようになる。

【0060】

バスリセットが発生した後、各ノードの接続状況を認識するために、各ノードの直結されているポート間において、親子関係の宣言がなされる。ここでいう親子とは、階層構造の上位が「親」、下位が「子」という意味である。図10では、バスリセットの後、最初に親子関係を宣言したのはノードAである。前述したように、一つのポートだけが接続されたノード（リーフ）から親子関係の宣言を開始することができる。これは、ポート数が「1」であればネットワークツリーの末端、つまりリーフノードであることが認識され、それらリーフノードの中で最も早く動作を行ったノードから親子関係が決定されて行くことになる。こうして親子関係の宣言を行ったノードのポートが、互いに接続された二つのノードの「子」と設定され、相手ノードのノードが「親」と設定される。こうして、ノードA-B間、ノードE-D間、ノードF-D間で「子-親」の関係が設定される。

【0061】

さらに、階層が一つ上がって、複数のポートをもつノード、つまりブランチノードのうち他ノードから親子関係の宣言を受けたノードから順次、上位のノードに対して親子関係を宣言する。図10ではまずノードD-E間、D-F間の親子関係が決定された後、ノードDがノードCに対して親子関係を宣言し、その結果、ノードD-C間で「子-親」の関係が設定される。ノードDから親子関係の宣言を受けたノードCは、もう一つのポートに接続されているノードBに対して親子関係を宣言し、これによってノードC-B間で「子-親」の関係が設定される。

【0062】

このようにして、図10に示すような階層構造が構成され、最終的に接続されているすべてのポートにおいて親となったノードBが、ルートノードと決定される。なお、ルートノードは一つのネットワーク構成中に一つしか存在しない。また、ノードAから親子関係を宣言されたノードBが、速やかに、他のノードに対して親子関係を宣言した場合は、例えばノードCなどの他のノードがルートノードに

なる可能性もあり得る。すなわち、親子関係の宣言が伝達されるタイミングによっては、どのノードもルートノードになる可能性があり、ネットワーク構成が同一であっても、特定のノードがルートノードになるとは限らない。

【0063】

ルートノードが決定されると、各ノードIDの決定モードに入る。すべてのノードは、決定した自分のID情報を、他のすべてのノードに通知するブロードキャスト機能をもっている。なお、ID情報は、ノード番号、接続されている位置の情報、もっているポートの数、接続のあるポートの数、各ポートの親子関係の情報などを含むID情報としてブロードキャストされる。

【0064】

ノード番号の割当ては、前述したようにリーフノードから開始され、順に、ノード番号=0,1,2,…が割当てられる。そしてID情報のブロードキャストによって、そのノード番号は割当て済みであることが認識される。

【0065】

すべてのリーフノードがノード番号を取得し終わると、次はブランチノードへ移りリーフノードに続くノード番号が割当てられる。リーフノードと同様に、ノード番号が割当てられたブランチノードから順にID情報がブロードキャストされ、最後にルートノードが自己のID情報をブロードキャストする。従って、ルートノードは常に最大のノード番号を所有することになる。

【0066】

以上のようにして、階層構造全体のID設定が終わり、ネットワーク構成が構築され、バスの初期化作業が完了する。

【0067】

[ノード管理のための制御情報]

ノード管理を行うためのCSRアーキテクチャの基本的な機能として、図4に示したCSRコアがレジスタ上に存在する。それらレジスタの位置と機能を図11に示すが、図中のオフセットは0xFFFFF00000000からの相対位置である。

【0068】

CSRアーキテクチャでは、0xFFFFF0000200からシリアルバスに関するレジスタ

が配置されている。それらのレジスタの位置と機能を図12に示す。

【0069】

また、0xFFFFF0000800から始まる場所には、シリアルバスのノード資源に関する情報が配置されている。それらのレジスタの位置と機能を図13に示す。

【0070】

CSRアーキテクチャでは、各ノードの機能を表すためコンフィグレーションROMをもっているが、このROMには最小形式と一般形式があり、0xFFFFF0000400から配置される。最小形式では図14に示すようにベンダIDを表すだけであり、このベンダIDは24ビットで表される全世界で固有の値である。

【0071】

また、一般形式は図15に示すような形式で、ノードに関する情報をもっているが、この場合、ベンダIDはルートディレクトリ(root_directory)にもつことができる。また、バス情報ブロック(bus info block)とルートリーフ(root leaf)にはベンダIDを含む64ビットの全世界で固有な装置番号をもっている。この装置番号は、バスリセットなどの再構成後に継続してノードを認識するために使用される。

【0072】

[シリアルバス管理]

1394シリアルバスのプロトコルは、図3に示したように、フィジカルレイヤ811、リンクレイヤ812およびトランザクションレイヤ814から構成されている。この中で、バス管理は、CSRアーキテクチャに基づくノードの制御とバス資源管理のための基本的な機能を提供している。

【0073】

バス管理を行うノード（以下「バス管理ノード」と呼ぶ）は、同一バス上に唯一存在し、シリアルバス上の他のノードに管理機能を提供するが、この管理機能にはサイクルマスターの制御や、性能の最適化、電源管理、伝送速度管理、構成管理などがある。

【0074】

バス管理機能は、バスマネージャ、同期（アイソクロノス）リソースマネージ

やおよびノード制御の三つの機能に大きく別けられる。ノード制御は、CSRによってフィジカルレイヤ811、リンクレイヤ812、トランザクションレイヤ814およびアプリケーションにおけるノード間通信を可能にする管理機能である。同期リソースマネージャは、シリアルバス上で同期型のデータ転送を行うために必要になる管理機能で、同期データの転送帯域幅とチャネル番号の割当てを管理するものである。この管理を行うためにバス管理ノードは、バスの初期化後に、同期リソースマネージャ機能をもつノードの中から動的に選出される。

【0075】

また、バス上にバス管理ノードが存在しない構成では、電源管理やサイクルマスターの制御のようなバス管理の一部の機能を同期リソースマネージャ機能をもつノードが行う。さらにバス管理は、アプリケーションに対してバス制御のインターフェイスを提供するサービスを行う管理機能であり、その制御インターフェイスにはシリアルバス制御要求(SB_CONTROL.request)、シリアルバスイベント制御確認(SB_CONTROL.confirmation)、シリアルバスイベント通知(SB_EVENT.indication)がある。

【0076】

シリアルバス制御要求は、バスのリセット、バスの初期化、バスの状態情報などを、アプリケーションからバス管理ノードに要求する場合に利用される。シリアルバスイベント制御確認は、シリアルバス制御要求の結果で、バス管理ノードからアプリケーションに確認通知される。シリアルバスイベント通知は、バス管理ノードからアプリケーションに対して、非同期に発生されるイベントを通知するためのものである。

【0077】

【データ転送プロトコル】

1394シリアルバスのデータ転送は、周期的に送信する必要のある同期データ（同期パケット）と、任意タイミングのデータ送受信が許容される非同期データ（非同期パケット）とが同時に存在し、なおかつ、同期データのリアルタイム性を保証している。データ転送では、転送に先立ってバス使用権を要求し、バスの使用許可を得るためのバスアービトリエーションが行われる。

【0078】

非同期転送においては、送信ノードIDおよび受信ノードIDが転送データと一緒にパケットデータとして送られる。受信ノードは、自分のノードIDを確認してパケットを受取るとアクノリッジ信号を送信ノードに返すことで、一つのトランザクションが完了する。

【0079】

同期転送においては、送信ノードが伝送速度とともに同期チャネルを要求し、チャネルIDが転送データと一緒にパケットデータとして送られる。受信ノードは、所望するチャネルIDを確認してデータパケットを受取る。必要になるチャネル数と伝送速度はアプリケーションレイヤ816で決定される。

【0080】

これらのデータ転送プロトコルは、フィジカルレイヤ811、リンクレイヤ812およびトランザクションレイヤ814の三つのレイヤによって定義される。フィジカルレイヤ811は、バスとの物理的・電気的インターフェイス、ノード接続の自動認識、ノード間のバス使用権のバスアービトリエーションなどを行う。リンクレイヤ812は、アドレッシング、データチェック、パケット送受信、そして同期転送のためのサイクル制御を行う。トランザクションレイヤ814は、非同期データに関する処理を行う。以下、各レイヤにおける処理について説明する。

【0081】

[フィジカルレイヤ]

次に、フィジカルレイヤ811におけるバスアービトリエーションを説明する。

【0082】

1394シリアルバスは、データ転送に先立って、必ず、バス使用権のアービトリエーションを行う。1394シリアルバスに接続された各機器は、ネットワーク上を転送される信号をそれぞれ中継することによって、ネットワーク内のすべての機器に同信号を伝える論理的なバス型ネットワークを構成するので、パケットの衝突を防ぐ意味でバスアービトリエーションが必要である。これによって、ある時間には、一つのノードだけが転送を行うことができる。

【0083】

図16はバス使用権の要求を説明する図、図17はバス使用の許可を説明する図である。バスアービトレーションが始まると、一つもしくは複数のノードが親ノードに向かって、それぞれバスの使用権を要求する。図16においては、ノードCとノードFがバス使用権を要求している。この要求を受けた親ノード（図16ではノードA）は、さらに親ノードに向かって、バスの使用権を要求することで、ノードFによるバスの使用権の要求を中継する。この要求は最終的に、アービトレーションを行うルートノードに届けられる。

【0084】

バスの使用権の要求を受けたルートノードは、どのノードにバスの使用権を与えるかを決める。このアービトレーション作業はルートノードのみが行えるものであり、アービトレーションに勝ったノードにはバスの使用許可が与えられる。図17は、ノードCにバスの使用許可が与えられ、ノードFのバスの使用権の要求は拒否された状態を示している。

【0085】

ルートノードは、バスアービトレーションに負けたノードに対してはDP(data prefix)パケットを送り、そのバスの使用権の要求が拒否されたことを知らせる。バスアービトレーションに負けたノードのバスの使用権の要求は、次回のバスアービトレーションまで待たされることになる。

【0086】

以上のようにして、アービトレーションに勝ってバス使用の許可を得たノードは、以降、データ転送を開始することができる。ここで、バスアービトレーションの一連の流れを図18に示すフローチャートにより説明する。

【0087】

ノードがデータ転送を開始できるためには、バスがアイドル状態であることが必要である。先に開始されたデータ転送が終了し、現在、バスがアイドル状態にあることを確認するためには、各転送モードで個別に設定されている所定のアイドル時間ギャップ長（例えば、サブアクションギャップ）の経過を検出し、所定のギャップ長が検出された場合、各ノードはバスがアイドル状態になったと判断する。各ノードは、ステップS401で、非同期データ、同期データなどそれぞれ転

送するデータに応じた所定のギャップ長が検出されたか否かを判断する。所定のギャップ長が検出されない限り、転送を開始するために必要なバス使用権を要求することはできない。

【0088】

各ノードは、ステップS401で所定のギャップ長が検出されると、ステップS402で転送すべきデータがあるか判断し、ある場合はステップS403でバスの使用権を要求する信号をルートに対して発信する。このバスの使用権の要求を表す信号は、図16に示すように、ネットワーク内の各機器に中継されながら、最終的にルートノードに届けられる。ステップS402で転送するデータがないと判断した場合は、ステップS401に戻る。

【0089】

ルートノードは、ステップS404でバスの使用権を要求する信号を一つ以上受信したら、ステップS405で使用権を要求したノードの数を調べる。ステップS405の判定により、使用権を要求したノードが一つだったら、そのノードに、直後のバス使用許可が与えられることになる。また、使用権を要求したノードが複数だったら、ステップS406で直後のバス使用許可を与えるノードを一つに絞るアビトレーション作業が行われる。このアビトレーション作業は、毎回同じノードばかりにバスの使用許可を与えるようなことはなく、平等にバスの使用許可を与えるようになっている（フェア・アビトレーション）。

【0090】

ルートノードの処理は、ステップS407で、ステップS406のアビトレーションに勝った一つのノードと、敗れたその他のノードとに応じて分岐する。アビトレーションに勝った一つのノード、またはバスの使用権を要求したノードが一つの場合は、ステップS408でそのノードに対してバスの使用許可を示す許可号が送られる。この許可信号を受信したノードは、直後に転送すべきデータ（パケット）の転送を開始する（ステップS410）。また、アビトレーションに敗れたノードにはステップS409で、バス使用権の要求が拒否されたことを示すDP(data prefix)パケットが送られる。DPパケットを受取ったノードの処理は、再度、バスの使用権を要求するためにステップS401まで戻る。ステップS410におけるデータの転

送が完了したノードの処理もステップS401へ戻る。

【0091】

[トランザクションレイヤ]

トランザクションの種類には、リードトランザクション、ライトトランザクションおよびロックトランザクションの三種類がある。

【0092】

リードトランザクションでは、イニシエータ（要求ノード）がターゲット（レスポンスノード）のメモリの特定アドレスからデータを読取る。ライトトランザクションでは、イニシエータがターゲットのメモリの特定アドレスにデータを書込む。また、ロックトランザクションでは、イニシエータからターゲットに参照データと更新データを転送する。その参照データは、ターゲットのアドレスのデータと組み合わされて、ターゲットの特定のアドレスを指示する指定アドレスになる。そして、この指定アドレスのデータが更新データにより更新される。

【0093】

図19はトランザクションレイヤ814におけるCSRアーキテクチャに基づくリード、ライト、ロックの各コマンドの要求・レスポンスプロトコルを示す図で、図に示す要求、通知、レスポンスおよび確認は、トランザクションレイヤ814でのサービス単位である。

【0094】

トランザクション要求(TR_DATA.request)はレスポンスノードに対するパケットの転送、トランザクション通知(TR_DATA.indication)はレスポンスノードに要求が届いたことの通知、トランザクションレスポンス(TR_DATA.response)はアクノリッジの送信、トランザクション確認(TR_DATA.confirmation)はアクノリッジの受信である。

【0095】

[リンクレイヤ]

図20はリンクレイヤ812におけるサービスを示す図で、レスポンスノードに対するパケットの転送を要求するリンク要求(LK_DATA.request)、レスポンスノードにパケット受信を通知するリンク通知(LK_DATA.indication)、レスポンスノー

ドからのアクノリッジ送信のリンクレスポンス(LK_DATA.response)、要求ノードのアクノリッジ送信のリンク確認(LK_DATA.confirmation)のサービス単位に分けられる。一つのパケット転送プロセスはサブアクションと呼ばれ、非同期サブアクションと同期サブアクションの二つの種類がある。以下では、各サブアクションの動作について説明する。

【0096】

[非同期サブアクション]

非同期サブアクションは非同期データ転送である。図21は非同期転送における時間的な遷移を示す図である。図21に示す最初のサブアクションギャップは、バスのアイドル状態を示すものである。このアイドル時間が所定値になった時点で、データ転送を希望するノードがバス使用権を要求し、バスアービトリエーションが実行される。

【0097】

バスアービトリエーションによりバスの使用が許可されると、次に、データがパケット転送され、このデータを受信したノードは、ACKギャップという短いギャップの後、受信確認用返送コードACKを返してレスポンスするか、レスポンスパケットを返送することでデータ転送が完了する。ACKは4ビットの情報と4ビットのチェックサムからなり、成功、ビジー状態またはペンドィング状態であることを示す情報を含み、すぐにデータ送信元のノードに返される。

【0098】

図22は非同期転送用パケットのフォーマットを示す図である。パケットには、データ部および誤り訂正用のデータCRCのほかにヘッダ部があり、そのヘッダ部には目的ノードID、ソースノードID、転送データ長や各種コードなどが書き込まれている。

【0099】

また、非同期転送は送信ノードから受信ノードへの一対一の通信である。送信元ノードから送り出されたパケットは、ネットワーク中の各ノードに行き渡るが、各ノードは自分宛てのパケット以外は無視するので、宛先に指定されたノードだけがそのパケットを受取ることになる。

【0100】

[スプリットトランザクション]

トランザクションレイヤ814におけるサービスは、図19で示したトランザクション要求およびトランザクションレスポンスのセットで行われる。ここで、ターゲット（レスポンスノード）のリンクレイヤ812およびトランザクションレイヤ814における処理が充分高速であれば、要求とレスポンスをリンクレイヤ812のそれぞれ独立したサブアクションで処理せず、一つのサブアクションで処理することが可能になる。しかし、ターゲットの処理速度が遅い場合は、要求とレスポンスを個別のトランザクションで処理する必要がある。そして、この動作をスプリットトランザクションと呼ぶ。

【0101】

図23はスプリットトランザクションの動作例を示す図で、イニシエータ（要求ノード）のコントローラからのライト要求に対して、ターゲットはペンドィングを返す。これにより、ターゲットは、コントローラのライト要求に対する確認情報を返すことができ、データを処理するための時間を稼ぐことができる。そして、データ処理に充分な時間が経過した後、ターゲットは、ライトレスポンスをコントローラに通知してライトトランザクションを終了させる。なお、このときの要求とレスポンスのサブアクションの間には、他のノードによるリンクレイヤ812の操作が可能である。

【0102】

図24はスプリットトランザクションを行う場合の転送状態の時間的遷移例を示す図で、サブアクション1は要求サブアクションを、サブアクション2はレスポンスサブアクションをそれぞれ表している。

【0103】

サブアクション1で、イニシエータはライト要求を表すデータパケットをターゲットに送り、これを受けたターゲットはアクノリッジパケットにより上記の確認情報を示すペンドィングを返すことで要求サブアクションが終了する。

【0104】

そして、サブアクションギャップが挿入された後、サブアクション2で、ター

ゲットはデータパケットが無データであるライトレスポンスを送り、これを受けたイニシエータはアクノリッジパケットでコンプリートレスポンスを返すことによってレスポンスサブアクションが終了する。

【0105】

なお、サブアクション1の終了からサブアクション2の開始に至る時間は、最小はサブアクションギャップに相当する時間であり、最大はノードに設定された最大待ち時間まで伸ばすことが可能である。

【0106】

【同期サブアクション】

1394シリアルバスの最大の特徴であるともいえるこの同期転送は、とくにAVデータなどのリアルタイム転送を必要とするデータの転送に適している。また、非同期転送が一対一の転送であるのに対し、この非同期転送はブロードキャスト機能によって、一つの送信元ノードから他のすべてのノードへ一様にデータを転送することができる。

【0107】

図25は同期転送における時間的な遷移を示す図で、同期転送はバス上で一定時間毎に実行され、この時間間隔を同期サイクルと呼ぶ。同期サイクル時間は125μSである。この同期サイクルの開始を示し、各ノードの動作を同期させる役割を担っているのがサイクルスタートパケット(CSP)2000である。CSP2000を送信するのは、サイクルマスターと呼ばれるノードであり、一つ前のサイクル内の転送が終了し、所定のアイドル期間(サブアクションギャップ2001)を経た後、本サイクルの開始を告げるCSP2000を送信する。つまり、このCSP2000が送信される時間間隔が125μSになる。

【0108】

また、図25にチャネルA、チャネルBおよびチャネルCと示すように、一つの同期サイクル内において複数種のパケットにチャネルIDをそれぞれ与えることにより、それらのパケットを区別して転送することができる。これにより、複数ノード間で、略同時に、リアルタイム転送が可能であり、また、受信ノードは所望するチャネルIDのデータのみを受信すればよい。このチャネルIDは、受信ノードの

アドレスなどを表すものではなく、データに対する論理的な番号に過ぎない。従って、送信されたあるパケットは、一つの送信元ノードから他のすべてのノードに行き渡る、つまりブロードキャストされることになる。

【0109】

同期転送によるパケット送信に先立ち、非同期転送と同様に、バスアービトリーシヨンが行われる。しかし、非同期転送のように一対一の通信ではないので、同期転送には受信確認用の返送コードのACKは存在しない。

【0110】

また、図25に示したisoギャップ（同期ギャップ）は、同期転送を行う前にバスがアイドル状態であることを確認するために必要なアイドル期間を表している。この所定のアイドル期間を検出したノードは、バスがアイドル状態にあると判断し、同期転送を行いたい場合はバス使用権を要求するのでバスアービトリーションが行われることになる。

【0111】

図26は同期転送用のパケットフォーマット例を示す図である。各チャネルに分けられた各種のパケットには、それぞれデータ部および誤り訂正用のデータCRCのほかにヘッダ部があり、そのヘッダ部には図27に示すような、転送データ長、チャネル番号、その他各種コードおよび誤り訂正用のヘッダCRCなどが書かれている。

【0112】

[バス・サイクル]

実際に、1394シリアルバスにおいては、同期転送と非同期転送が混在できる。図28は同期転送と非同期転送が混在するときの転送状態の時間的遷移を示す図である。

【0113】

ここで、前述したように同期転送は非同期転送より優先して実行される。その理由は、CSPの後、非同期転送を起動するために必要なアイドル期間のギャップ（サブアクションギャップ）よりも短いギャップ（アイソクロナスギャップ）で、同期転送を起動できるからである。従って、非同期転送より同期転送は優先し

て実行されることになる。

【0114】

図28に示す一般的なバスサイクルにおいて、サイクル# m のスタート時にCSPがサイクルマスターから各ノードに転送される。CSPによって、各ノードの動作が同期され、所定のアイドル期間（同期ギャップ）を待ってから同期転送を行おうとするノードはバスアービトレーションに参加し、パケット転送に入る。図28ではチャネルe、チャネルsおよびチャネルkが順に同期転送されている。

【0115】

このバスアービトレーションからパケット転送までの動作を、与えられているチャネル分繰り返し行った後、サイクル# m における同期転送がすべて終了すると、非同期転送を行うことができるようになる。つまり、アイドル時間が、非同期転送が可能なサブアクションギャップに達することによって、非同期転送を行いたいノードはバスアービトレーションに参加する。ただし、非同期転送が行えるのは、同期転送の終了から、次のCSPを転送すべき時間(cycle sync)までの間に、非同期転送を起動するためのサブアクションギャップが検出された場合に限られる。

【0116】

図28に示すサイクル# m では、三つのチャネル分の同期転送の後、非同期転送によりACKを含む2パケット（パケット1、パケット2）が転送されている。この非同期パケット2の後、サイクル $m+1$ をスタートすべき時間(cycle sync)に至るので、サイクル# m における転送はこれで終わる。ただし、非同期または同期転送中に次のCSPを送信すべき時間(cycle sync)に至ったら、転送を無理に中断せず、その転送が終了した後にアイドル期間を経て次の同期サイクルのCSPを送信する。すなわち、一つの同期サイクルが $125\ \mu s$ 以上続いたときは、その延長分、次の同期サイクルは基準の $125\ \mu s$ より短縮される。このように同期サイクルは $125\ \mu s$ を基準に超過、短縮し得るものである。

【0117】

しかし、同期転送はリアルタイム転送を維持するために、必要であれば毎サイクル実行され、非同期転送は同期サイクル時間が短縮されたことによって次以降

の同期サイクルに延期されることもある。サイクルマスタは、こういった遅延情報も管理する。

【0118】

[FCP]

AV/Cプロトコルでは、1394シリアルバス上の装置を制御するために機能制御プロトコルFCP(Functional Control Protocol)が用意されている。FCPの制御コマンドの送信とレスポンスには、IEEE1394で規定されている非同期パケットが用いられる。FCPにおいては、制御側のノードをコントローラ、被制御側のノードをターゲットと呼び、コントローラからターゲットに送られるFCPパケットフレームをAV/Cコマンドフレーム、逆にターゲットからコントローラに返されるFCPパケットフレームをAV/Cレスポンスフレームと呼ぶ。

【0119】

図29は、ノードAがコントローラ、ノードBがターゲットの場合を示し、それぞれに用意されているレジスタアドレスのうち0x0000B00番地からの512バイトがコマンドレジスタ、0x0000D00番地からの512バイトがレスポンスレジスタであり、それぞれ非同期転送を用いたパケットフレームにより、指定されたアドレスのレジスタにデータが書き込まれる。このときのコントローラによるAV/Cコマンドフレームの送信と、ターゲットによるAV/Cレスポンスフレームのレスポンスの関係は、AV/Cトランザクションと呼ばれる。一般的なAV/Cトランザクションでは、ターゲットはコマンドフレームを受取ってから100ms以内に、コントローラに対してレスポンスフレームをレスポンスする必要がある。

【0120】

図30はFCPパケットフレームを含む非同期転送のパケットフォーマットを示す図で、図22に示した非同期データパケットのデータ領域に対して、コマンドフレームやレスポンスフレームを挿入してAV/Cトランザクションが実行される。

【0121】

図31はAV/Cコマンドフレームの構造を示す図、図32はAV/Cレスポンスフレームの構造を示す図で、FCPパケットフレームとしてはヘッダのctype、response、subunit_type、subunit_IDの後に、FCPのデータ部分がつながった構造になってい

る。

【0122】

`ctype`は、コマンドフレームにおけるコマンドタイプを示し、CONTROL、STATUS、INQUIRY、NOTIFYの各状態を示している。

【0123】

`response`は、レスポンスフレームにおけるレスポンスコードを示し、ACCEPTED、REJECTED、IN_TRANSITION、IMPLEMENTED、CHANGED、INTERIMなどの各状態を示している。

【0124】

また、`subunit_type`はデバイスの分類、`subunit_ID`はインスタンス番号を示している。

【0125】

FCPのデータ部分はオペコード+オペランドの構成になっていて、各種のAV/Cコマンドを使ってターゲットの制御を行ったり、AV/Cレスポンスのレスポンスをすることができる。

【0126】

コマンドフレームにおけるオペコード(`opcode`)は、図50に示す`command`列に示されたコマンド群の一つであり、各コマンドは`ctype`にセットされるコマンドタイプに従った内容で実行される。

【0127】

`ctype`により「CONTROL」が指定されたコマンドは、制御コマンドを意味し、ターゲットの機器を制御したり、オペランド(`operand`)以降に設定された内容でターゲットを設定する。`ctype`により「STATUS」が指定されたコマンドは、そのコマンドに対応するステータスを得るためにものである。`ctype`により「INQUIRY」が指定されたコマンドは、そのコマンドにより設定可能な内容を問い合わせるためにものである。`ctype`に「NOTIFY」が指定されたコマンドは、そのコマンドの確認を行いうためのものである。

【0128】

各コマンドは、各コマンドに必要な内容がオペランドに設定され、コマンドフ

レームに書込まれる。

【0129】

レスポンスフレームにおけるオペコードには、図50のresponse列に示されるレスポンスコードが設定される。各レスポンスは、各レスポンスに対応するオペランドを有する。レスポンスのオペランドには、コマンドの実行が正常に終了したか、エラー終了したかを示す情報がセットされるので、このオペランドに従ってエラー処理を行うことができる。

【0130】

【LOGINプロトコル】

【LOGINプロトコルを用いた通信】

図33は、1394シリアルバスのインターフェイスをLANでよく用いられるOSIモデルの各層と対比させた図である。OSIモデルの物理層1とデータリンク層2が、1394シリアルバスのインターフェイスの下位層4であるフィジカルレイヤ811およびリンクレイヤ812に該当する。下位層4の上に存在する1394シリアルバスのインターフェイスにおけるトランスポートプロトコル層5とプレゼンテーション層6は、OSIモデルのネットワーク層、トランスポート層、セッション層およびプレゼンテーション層を含む上位層3に該当する。また、LOGINプロトコル7は、1394シリアルバスのインターフェイスの下位層4とトランスポートプロトコル層5との間で動作するものである。

【0131】

図33に示す例1では、プリンタなどの周辺機器用のシリアルバスプロトコル(SBP-2)8に準拠したデバイスにLOGINプロトコル7をもたせることによって、相手のデバイスに対してSBP-2に準拠したプロトコルを使ってデータのやり取りを行いたいことを通知させることができる。

【0132】

また図33に示す例2では、1394シリアルバスのインターフェイス上で特化されたデバイスプロトコル9についても、LOGINプロトコル7をもたせることで、デバイスが互いに、互いのプロトコルをサポートしているかを判別させることができる。

【0133】

図34はLOGINプロトコル7の基本動作を示す図で、プリンタデバイスは、ホストデバイスからの印刷タスク(print task)10を実行する際に、まず、プリンタに用意されているプリンタプロトコルA、BおよびCの内、どれを選択して印刷データをやり取りするかをLOGINプロトコル7による通信に基づき決定し、その後は、決定したプリンタプロトコルに従って印刷データのやり取りを行う。すなわち、幾つかのプリンタプロトコルをサポートしているプリンタデバイスは、ホストデバイスと接続する際に、まずホストデバイスに用意されているトランスポートプロトコル5をLOGINプロトコル7によって判別し、ホストデバイスのトランスポートプロトコル5に合ったプリンタプロトコルを選択し、選んだプリンタプロトコルに従って印刷データやコマンドのやり取りを行うことで、印刷タスク10の処理を行う。

【0134】

図35は1394シリアルバスにおける接続形態の一例を示す図で、複数のプリンタプロトコルに対応(multi-protocol support)するプリンタ11に対してLOGINプロトコル7を実装したデバイス(PC12、スキャナ13、DVC14等)が接続された状態を示している。プリンタ11は、LOGINプロトコル7により判別した、接続を要求する相手デバイスのトランスポートプロトコル5に応じてプリンタプロトコルを切替えることにより、各デバイスからの印刷タスクを問題なく処理することが可能となる。

【0135】

図36はログイン動作の流れを示す図である。ステップ1において：ホストデバイスは、ターゲットデバイス（この場合マルチプロトコルプリンタ）をロックする。ターゲットデバイスは、ホストデバイスのケーパビリティ（トランスポートプロトコルを含む）を調べ、かかるケーパビリティは、後述するレジスタ503に格納される。

【0136】

ステップ2において：ステップ1で設定されたプロトコルによりプリントデータの通信が行われる。

【0137】

ステップ3において：ホストデバイスは、ターゲットデバイスとのコネクションを切断する。

【0138】

図37はLOGINプロトコル7のためにターゲットデバイスであるプリンタが備える1394シリアルバスのCSRを示し、ロックレジスタ(lock)501、プロトコルレジスタ(protocol)502、ケーパビリティレジスタ(capability)503を示している。これらのレジスタは1394シリアルバスのアドレス空間における初期ユニット空間の定められたアドレスに配置される。つまり、図4を用いて説明したように、機器に与えられたアドレス幅48bitのうち、最初の20ビットにおける0xFFFFがレジスタ空間と呼ばれ、その最初の512バイトにCSRアーキテクチャのコアになるレジスタ(CSRコア)が配置されている。

【0139】

ロックレジスタ501は、リソースのロック状態を示し、値「0」はログイン可能な状態を表し、「0」以外はロック状態で既にログインされていることを表す。ケーパビリティレジスタ503は、そのビット毎に設定可能なプロトコルを示し、値‘1’のビットに対応するプロトコルは設定可能であることを表し、‘0’に対応するプロトコルは設定不可能であることを表す。プロトコルレジスタ502は、現在設定されているプロトコルを示し、設定されたプロトコルに対応するケーパビリティレジスタ503のビットに相当するビットの値が‘1’になる。図38はホストデバイスにおけるログイン処理を示すフローチャートである。

【0140】

ログインを開始するためには、まずログインしようとするターゲットデバイス、例えばプリンタのロックレジスタ501、プロトコルレジスタ502およびケーパビリティレジスタ503のデータをリードトランザクションにより確認する。ここで、ケーパビリティレジスタ503のデータから、ホストデバイスが通信に用いるプロトコルをターゲットデバイスがサポートしているかどうか確認する(ステップS601)。

【0141】

もしホストデバイスのプロトコルがターゲットデバイスのサポート外ならば、次のステップS602でログインを中止する。また、ロックレジスタ501のデータが「0」以外であれば、他のデバイスがログイン中であるとみなしログインを中止する。ログインレジスタ501のデータが「0」であれば現在ログイン可能とみなす（ステップS602）。

【0142】

ログイン可能の場合、リソースロック処理に移り、プリンタのロックレジスタ501にロックトランザクションを用いて例えば「1」を書き込みログインを設定する（ステップS603）。この状態でターゲットデバイスはロックされたことになり、他のデバイスからの制御は不可、またレジスタの変更も不可能となる。

【0143】

上記のようにターゲットデバイスのリソースがロックされた状態で、次にプロトコルの設定を行う。ターゲットデバイスである本実施例のプリンタは、複数のプリンタプロトコルをサポートするため、プリントデータを受取る前に、ホストデバイスが使用できるプロトコルを知らねばならない。本実施例においては、ホストデバイスのライトトランザクションにより、プリンタのプロトコルレジスタ502の相当するビットを設定することで、これから使用するプロトコルをプリンタに通知する（ステップS604）。

【0144】

この時点で、ホストデバイスが通信に用いるプロトコルがターゲットデバイスに通知され、かつターゲットデバイスがロック状態なので、現在、ターゲットデバイスにログインしているホストデバイスがデータ、この場合はプリントデータの送信を行う（ステップS605）。

【0145】

データの送信が終了したら、ホストデバイスはターゲットデバイスのロックレジスタ501およびプロトコルレジスタ503をクリアすることにより、プリンタからログアウトする（ステップS606）。

【0146】

図39はターゲットデバイスであるプリンタのログイン処理を示すフローチャー

トである。プリンタは、通常、ホストデバイスからログインされるのを待つ状態にある。ホストデバイスからのプリントリクエストは、プリンタのロックレジスタ501、プロトコルレジスタ502およびケーパビリティレジスタ503の読み取りにより開始されるので、前記レジスタは、常に他のデバイスから読み出し可能な状態にしておく必要がある。

【0147】

今、プリントアウトを実行しようとするホストデバイスによりプリンタがロックされたとする（ステップS701）。プリンタは、次にホストデバイスから使用プロトコルが通知されるのを待つ（ステップS702）。プリンタが、ロック状態になってから使用プロトコルの通知を待つのは、ログインの途中で、他のデバイスからのリクエストにより、プロトコルレジスタ503を書き換えられないようするためである。

【0148】

使用プロトコルの通知があったら（ステップS703）、プリンタは、通知された使用プロトコルに自分のプロトコルをスイッチして（ステップS704、S706およびS708）、ホストデバイスのプロトコルに合わせて通信を行う（ステップS705、S707またはS709）。

【0149】

通信が終了したら、プリンタはロックレジスタ501およびプロトコルレジスタ503がクリアされたのを確認し（ステップS710）、ログインを待つ状態（ステップS701）に戻る。

【0150】

[LOGINプロトコルをもたないデバイスを考慮した例]

図40はLOGINプロトコル7を実装していないデバイスを考慮した例を説明する図で、図34に示した第一例と比較すると、LOGINプロトコル7をもたずプロトコルDをもつデバイスについても対応している点が特徴である。すなわち、LOGINプロトコル7をもつデバイスだけでなく、既存のプロトコルD（例えばAV/Cプロトコル）にのみ対応しているデバイスに対しても例えば印刷動作を保証するために、プリンタ側にLOGINプロトコル7をもたないデバイスに対応するプリンタプロトコル

を追加したものである。

【0151】

この動作について説明すると、接続の初めに行われるプリントリクエストによってホストデバイスがLOGINプロトコル7に対応していないことをプリンタが認識した場合、プリンタはプロトコルDを使ってホストデバイスとの通信を試み、通信が成立した場合はそのプロトコルDに従って印刷タスク10を実行する。

【0152】

図41は、図40に示した構成とOSIモデルの各層とを対比させた図で、例3は、LOGINプロトコル7が実装されていない現行のAV/Cプロトコルに準拠したAVデバイス15をモデルとしている。例4は、LOGINプロトコル7が実装されていないスキャナ用の非標準プロトコルが実装されているスキャナ16をモデルとしている。

【0153】

すなわち、LOGINプロトコル7を実装していないプロトコルをもつデバイスについても、そのデバイスがもつプロトコルにプリンタが対応することができれば、そのプリンタを利用することができるデバイスの種類を拡げることができる。

【0154】

【ダイレクトプリントプロトコル】

以下では、本発明にかかるプリンタと画像供給デバイスとの印刷手順を説明するが、プリンタと画像供給デバイスとを直結し、画像供給デバイスから供給される画像データに基づく画像形成をプリンタに行わせるためのプロトコルを「ダイレクトプリントプロトコル(DPP)」と呼ぶ。

【0155】

DPPの基本は、1394シリアルバスの初期ユニット空間（図4のユニット空間で示される）内にコマンド書込みのためのコマンドレジスタ(command)、コマンドに対するレスポンスを書込むためのレスポンスレジスタ(response)、転送データを書込むためのデータレジスタ(data)、転送データ個別のデータフォーマットに対応したフォーマット情報を扱うためのフォーマットレジスタ(format)から構成される。

【0156】

図42はこのレジスタマップを示す図で、コマンドレジスタ、レスポンスレジスタは図29を用いて説明したものと同じである。以降の説明では、画像供給デバイスをコントローラ、プリンタをターゲットとし、画像供給デバイスからプリンタへ画像データを供給して画像を印刷する例を説明する。

【0157】

コマンドレジスタ42-1は、プリンタ側の0xFFFFF0000B00の固定アドレスに配置され、512バイトのメモリ空間を有し、画像供給デバイスがプリンタに対する各種コマンドを書込むためのものである。勿論、コマンドレジスタが画像供給デバイス側にもある場合、プリンタからコマンドを書込むことも可能になる。コマンドレジスタ42-1に書込まれるコマンドをコマンドフレームという。

【0158】

レスポンスレジスタ42-2は、画像供給デバイス側の0xFFFFF0000D00の固定アドレスに配置され、512バイトのメモリ空間を有し、画像供給デバイスからプリンタに対して書込まれた各種コマンドに対するレスポンスが書込まれるためのものである。勿論、レスポンスレジスタがプリンタ側にもある場合、画像供給デバイスからレスポンスを書込むことも可能になる。レスポンスレジスタ42-2に書込まれるレスポンスをレスポンスフレームという。なお、図29においてはアドレスの上位0xFFFFが省略して記述されている。

【0159】

データレジスタ42-3は、FFFFF0003000hをデフォルトアドレスとし、BlockAddress, BufferConfigコマンド（データレジスタのアドレスを定義するコマンド）により有効な任意のアドレスに設定される。データレジスタ42-3のメモリ空間は、BlockSize, BufferConfigコマンド（データレジスタのメモリ空間を定義するコマンド）で前もって決められた範囲で設定される。データレジスタ42-3は、画像供給デバイスとプリンタの間でデータの転送を行うためのレジスタで、印刷を行う場合には画像供給デバイスからプリンタに印刷データが書込まれる。印刷データは、予め設定された画像フォーマットのデータ形式に従い、データレジスタ42-3に書込まれるデータをデータフレームという。

【0160】

フォーマットレジスタ42-4は、後述する各データフォーマットに対応したレジスタ群をまとめたもので、各レジスタは各データフォーマットに必要なフォーマット情報を設定するためのレジスタになっている。つまり、フォーマットレジスタ42-2は、画像供給デバイスからプリンタに対してフォーマット情報を書込むためのレジスタである。フォーマットレジスタ42-2に書込まれるフォーマット情報をフォーマットフレームという。

【0161】

図43は画像供給デバイスからプリンタへのフレームの流れを示す図で、コマンドフレーム、レスポンスフレーム、データフレーム、フォーマットフレームの流れを表し、画像供給デバイスから出力されたフレームに従いプリンタは印刷を行う。

【0162】

画像供給デバイスからプリンタへ送られるコマンドは、コマンドフレームとしてプリンタのコマンドレジスタ43-4へ書込まれる。このコマンドは、図50に示す印刷を行わせるためのコマンドである。このコマンドに対するレスポンスは、プリンタにより画像供給デバイスのレスポンスレジスタ43-2に書込まれる。このレスポンスは、コマンドを正しく実行されたかどうかを示す情報およびコマンドに対する戻り値などを含む。画像供給デバイスからプリンタへ送られるプリントデータは、データフレームとしてプリンタのデータレジスタ43-6へ書込まれる。画像供給デバイスからプリンタへ送られるフォーマット情報は、フォーマットフレームとしてプリンタのフォーマットレジスタ43-7に書込まれる。

【0163】

図44はフォーマットレジスタ43-7の構成例を示す図である。フォーマットレジスタ43-7は、問い合わせのための読込専用レジスタINQUIRY 44-1と、設定および情報取得のための読込/書込レジスタCONTROL/STATUS 44-2とに区分される。INQUIRY 44-1およびCONTROL/STATUS 44-2は、同じ構成のレジスタグループから構成される。つまり、レジスタグループを構成するレジスタ44-3から44-7と、レジスタ44-9から44-13である。

【0164】

さらに詳しく説明すると、フォーマットレジスタグループは、共通レジスタグループ(`print common register group`)を構成するレジスタ44-3および44-4(44-9および44-10)と、プリンタフォーマットレジスタグループ(`print format register group`)を構成するレジスタ44-5から44-7(44-11から44-13)とから構成される。

【0165】

共通レジスタグループは、すべてのデータフォーマットに共通なレジスタを集めたレジスタ群であり、すべてのプリンタに共通なレジスタGLOBAL 44-3(44-9)と、個別のプリンタに固有のレジスタLOCAL 44-4(44-10)とから構成される。

【0166】

プリンタフォーマットレジスタグループは、各データフォーマットに独自の情報を集めたレジスタ群であり、レジスタ`format[1] 44-5(44-11)`から`format[n] 44-7(44-13)`までのn個のレジスタ群からなる。レジスタ`format[1]`から`format[n]`は、後述するデータフォーマットに対応し、実装するデータフォーマットごとに一つのプリンタフォーマットレジスタグループが割り当てられる。

【0167】

なお、各フォーマットレジスタのアドレスは、データフォーマットを設定するコマンドに対するレスポンスとして画像供給デバイスに与えられる。

【0168】

図45は共通レジスタグループのステータスレジスタ`status register 44-8`の詳細な構成例を示す図である。ステータスレジスタ44-8はそれぞれ32ビットの、各ベンダのプリンタに共通なステータスを保持する共通ステータスレジスタ(`common status register`)45-1と、各ベンダにより定義されるステータスを保持するベンダ固有ステータス(`vendor specific status register`)45-2とから構成される。また、共通ステータスレジスタ45-1の第31ビットにあるvフラグ(`vendor status available flag`)により、ベンダ固有ステータスレジスタ45-2への拡張が次のように定義される。

vフラグ: '0' = `not available`

'1' = `available`

【0169】

また、共通ステータスレジスタ45-1に保持される情報は次のようなものがある

```

error.warning: エラー、ワーニングなどのステータス
paper state: 記録紙に関するステータス
print state: 印刷状況に関するステータス

```

【0170】

図46は共通レジスタグループのレジスタGLOBAL 44-3(44-9)に保持される情報の詳細例を示す図である。レジスタGLOBAL 44-3(44-9)は、DPP（ダイレクトプリントプロトコル）を搭載するプリンタすべてに共通な情報を保持する、つまり、プリンタの種別による差異がない情報を集めたレジスタである。図46はその一例を示し、記録媒体の種類を示す**media-type** 46-1、記録紙の大きさを示す**paper-size** 46-2、ページのマージン値を示す**page-margin** 46-3、ページの長さを示す**page-length** 46-4、ページのオフセットを示す**page-offset** 46-5、プリンタのユニット情報を示す**print-unit** 46-6、プリンタのカラータイプを示す**color-type** 46-7、データのビット順序を示す**bit-order** 46-8などを含む。

【0171】

図47は共通レジスタグループのレジスタLOCAL 44-4(44-10)に保持される情報の詳細例を示す図である。レジスタLOCAL 44-4(44-10)は、DPPを搭載するプリンタの機種それぞれに独自な情報を保持する、つまり、プリンタの種別により異なる情報を集めたレジスタである。図47はその一例を示し、プリンタ独自の記録紙の種類を示す**paper** 47-1、カラーマッチング方法を示すCMS 47-2、例えばインクジェットプリンタのインクの種類を示す**ink** 47-3などを含む。

【0172】

図48はレジスタ**format [1]** 44-5に保持される情報の一例を示す図で、画像データフォーマットの一つであるEXIF(Exchangeable Image File Format)に関する情報が保持された例である。この場合、入力のX方向の割合**inX-rate** 48-1、入力のY方向の割合**inY-rate** 48-2、出力のX方向の割合**outX-rate** 48-3、出力のY方向の割合**outY-rate** 48-4を含む。この場合、与えられたEXIFフォーマットの画像データ

タを各レジスタの内容に合わせてXY方向に変倍することにより印刷が可能になる。

【0173】

図49はレジスタformat[2] 44-6に保持される情報の一例を示す図で、画像データフォーマットの一つであるRaw RGBフォーマットに関する情報が保持された例である。Raw RGBフォーマットは、各ピクセルをR(red), G(green), B(blue)のデータで構成する画像フォーマットである。この場合、入力のX方向の割合inX-rate 49-1、入力のY方向の割合inY-rate 49-2、出力のX方向の割合outX-rate 49-3、出力のY方向の割合outY-rate 49-4、XYの固定ピクセルサイズ示すXY-size 49-5、ピクセル当りのビット数を示すbit-pixel 49-6、X方向のピクセル数を示すX-size 49-7、Y方向のピクセル数を示すY-size 49-8、色プレーン数を示すplane 49-9、X方向の解像度を示すX-resolution 49-10、Y方向の解像度を示すY-resolution 49-11、ピクセルの種類を示すpixel-format 49-12などを含む。この場合、与えられたRaw RGBフォーマットの画像データを各レジスタの内容に合わせてXY方向の変倍や解像度変換を行い、さらに画像サイズに関する情報などに基づいて処理することにより印刷することが可能になる。

【0174】

図50はコマンドおよびコマンドに対するレスポンスの一覧を示す図である。コマンドは幾つかの種類に分類される。分類としては、ステータス関係の「ステータス」、プリンタ制御のための「コントロール」、データ転送の設定のための「ブロック/バッファ」、チャネル設定を行う「チャネル」、転送方法に関する「転送」、フォーマット設定に関する「フォーマット」、ログインに関する「ログイン」およびデータ転送に関する「データ」などがある。

【0175】

さらに詳しく説明すれば、「ステータス」としてプリンタのステータスを得るためにコマンドGetStatusおよびそのレスポンスGetStatusresponse 50-1などがある。

【0176】

「コントロール」としては、プリンタのリセットを行うコマンドPrintResetお

およびそのレスポンスPrintResetResponse 50-2、印刷の開始を指示するPrintStartおよびそのレスポンスPrintStartResponse 50-3、印刷の中止を指示するPrintStopおよびそのレスポンスPrintStopResponse 50-4、記録紙の供給を指示するInsertPaperおよびそのレスポンスInsertPaperResponse 50-5、記録紙の排出を指示するEjectPaperおよびそのレスポンスEjectPaperResponse 50-6、画像データのコピー開始を指示するCopyStartおよびそのレスポンスCopyStartResponse 50-7、並びに、画像データのコピー終了を指示するCopyEndおよびそのレスポンスCopyEndResponse 50-8などがある。

【0177】

「ブロック/バッファ」としては、ブロックのサイズを指定するBlockSizeおよびそのレスポンスBlockSizeResponse 50-9、ブロックのアドレスを指定するBlockAddressおよびそのレスポンスBlockAddressResponse 50-10、空きブロックの数を取得するFreeBlockおよびそのレスポンスFreeBlockResponse 50-11、ブロックへのデータ書き込みを指示するWriteBlocksおよびそのレスポンスWriteBlocksResponse 50-12、バッファの情報を指定するBufferConfigおよびそのレスポンスBufferConfigResponse 50-13、並びに、バッファからのデータ取得の開始を指定するSetBufferおよびそのレスポンスSetBufferResponse 50-14などがある。

【0178】

「チャネル」としては、チャネルをオープンするOpenChannelおよびそのレスポンスOpenChannelResponse 50-15、並びに、チャネルをクローズするCloseChannelおよびそのレスポンスCloseChannelResponse 50-16などがある。

【0179】

「転送」としてはデータ転送方法を指定するTransferMethodおよびそのレスポンスTransferMethodResponse 50-17などがある。

【0180】

「フォーマット」としてはフォーマットを設定するSetFormatおよびそのレスポンスSetFormatResponse 50-18などがある。

【0181】

「ログイン」としては、ログインを行うLoginおよびそのレスポンスLoginResp

onse 50-19、ログアウトを行うLogoutおよびそのレスポンスLogoutResponse 50-20、並びに、リコネクションを行うReconnectおよびそのレスポンスReconnectResponse 50-21などがある。

【0182】

以上のコマンドはコマンドフレームに書込まれるものである。

【0183】

さらに、「データ」として、データを書込むためのWriteBlock 50-22、WriteBuffer 50-23、および、データを読込むためのPullBuffer 50-24などがある。これらのコマンドは、データフレームに対して書込み、読み込みを行うものであり、対応するレスポンスはない。

【0184】

画像供給デバイスは、図31に示したコマンドフレームのオペコード(opcode)に、図50に示す各コマンドに対応する値をセットし、プリンタのコマンドレジスタ43-4に書込むことにより、プリンタによってそのコマンドが実行される。コマンドに対するレスポンスはコマンドと等しい値をもつ。プリンタは、図32に示したレスポンスフレームのオペコードにレスポンスをセットし、画像供給デバイスのレスポンスレジスタ43-2に書込む。このレスポンスにより、画像供給デバイスは各コマンドの実行結果を受取ることができる。

【0185】

図51はDPPがサポートする画像データフォーマット例を示す図である。プリンタはこれらのフォーマットのうち、少なくとも一つのロー(RAW)画像データをサポートしなければならない。またオプションとして、他の複数のフォーマットをサポートすることができる。

【0186】

図52はフォーマット設定処理の流れを示す図である。まず、画像供給デバイスは、ステップS500でSetFormat(INQUIRY)コマンドをプリンタへ送り、プリンタはステップS501でSetFormatレスポンスを返す。返されたレスポンスにより、画像供給デバイスはプリンタのINQUIRYレジスタのアドレスを知る。

【0187】

次に、画像供給デバイスは、ステップS502でSetFormat(CONTROL/STATUS)コマンドをプリンタへ送り、プリンタはステップS503でSetFormatレスポンスを返す。返されたレスポンスにより、画像供給デバイスはプリンタのCONTROL/STATUSレジスタのアドレスを知る。

【0188】

画像供給デバイスは、ステップS504-1からS504-nでプリンタのINQUIRYレジスタを読み、プリンタがサポートするフォーマットおよびフォーマットの設定項目を知る。次に、画像供給デバイスは、ステップS505-1からS505-nでプリンタのSTATUS/CONTROLレジスタを読み、フォーマットの設定値を知るとともに、ステップS506-1からS506-nでプリンタのSTATUS/CONTROLレジスタにデータを書込み、フォーマットを設定する。

【0189】

DPPにおけるデータ転送には以下の二種類のパケットを用いる。

- ・ フロー制御のための制御コマンドパケット
- ・ データ送信のためのパケット

【0190】

本実施形態では、フロー制御、データ送信方法の違いにより、以下に五種類のデータ転送方式を列挙するが、何れの方式においてもフロー制御のための制御コマンドはFCPに準拠している。

転送方式1: レスポンスマネル(Response Model)

転送方式2: 簡易レスポンスマネル(Simplified Response Model)

転送方式3: プッシュラージバッファモデル(PUSH Large Buffer Mode

1)

転送方式4: プルバッファモデル(PULL Buffer Model)

転送方式5: 同期モデル(Isochronous Model)

【0191】

実際の転送においては上記の方式の一つを選択し設定するが、その手順は図52に示したフォーマット設定手順と同様である。なお、図53に示すように、コマンドにはTransferMethodを、レスポンスにはTransferMethodResponseを用いる。

【0192】

図53はデータ転送方式の設定手順の一例を示す図である。画像供給デバイスは、コマンドタイプINQUIRYのTransferMethodコマンドおよびレスポンスにより、現在使用可能なプリンタのデータ転送方式を取得し(S600-1およびS600-2)、コマンドタイプCONTROL/STATUSのTransferMethodコマンドおよびレスポンスにより、現在、プリンタに設定されているデータ転送方式を取得し、設定する(S600-3およびS600-4)。

【0193】

ところで、上記の五種類の何れの方式においても、フロー制御のための制御コマンドは、1394シリアルバス上の装置を制御するためのプロトコルであるFCPに準拠している。FCPの制御コマンドの転送は、送信-応答とも、常に非同期のライトトランザクションにより行われる。

【0194】

図54は前記転送方式1および転送方式2に必要なレジスタの1394シリアルバスのアドレス空間におけるレジスタマップを示す図である。本実施形態では、制御する側のノードを画像供給デバイス(FCPではコントローラに対応)に、制御される側のノードをプリンタ(FCPではターゲットに対応)にする。

【0195】

画像供給デバイス、プリンタとともに、レジスタ空間のオフセット0x0B00から512バイトのコマンドレジスタ(601-1および601-7)、オフセット0x0D00から512バイトのレスポンスレジスタ(601-2および601-8)をもつ。これらレジスタはFCPプロトコルおよびAV/Cコマンドに準拠している。

【0196】

フロー制御は、これらのコマンドレジスタ(601-1および601-7)およびレスポンスレジスタ(601-2および601-8)にコマンドフレーム601-4およびレスポンスフレーム601-5を書き込むことにより行われる。また、印刷データの転送については専用のデータフレームを定義する。つまり、レジスタ空間のオフセット0x300からブロックサイズ分のデータレジスタ(601-3および601-9)を設け、データレジスタ(601-3および601-9)にデータフレーム601-6を書き込むことで印刷デ

ータの転送が行われる。なお、ブロックサイズは例えば512バイトにする。

【0197】

図55はデータパケットフレームの一例を示す図で、データヘッダ602-1、データペイロード602-2などから構成される。

【0198】

図56はデータヘッダ602-1の構成例を示す図で、上位8ビットはブロックカウント領域603-1であり、下位8ビットは将来のための予約領域S603-2である。ブロックカウント領域603-1は、ターゲット（プリンタ）により内部的に用いられるもので、一回のブロック転送により転送されるブロックの数をカウントするカウンタである。なお、ブロックカウント領域603-1は8ビットなので0から255の値を取り得る。

【0199】

図57はブロック転送におけるプリンタ内のデータパケットフレーム処理を示す図である。プリンタに受信されたデータパケットは、まず、プリンタのデータレジスタ604-1に書き込まれる。プリンタはデータパケットと同じサイズのバッファ（604-2から604-5）を有し、データレジスタ604-1に書き込まれたデータパケットは、順次、これらのバッファ（604-2から604-5）に移される。なお、これらデータバッファの数は、ブロックカウント数の最大値に等しい255個あることが望ましいが、これ以下でもよい。ここで、各バッファはブロックカウント値に対応し、ブロックカウントが「0」のデータパケットはBlock[0]のバッファに、ブロックカウントが「1」のデータパケットはBlock[1]のバッファに、順次、格納される。バッファ（604-2から604-5）に格納されたデータパケットは、データヘッダ602-1が取除かれた上で、プリンタのメモリ空間604-6に展開される。

【0200】

[転送方式1]

Response Modelである転送方式1は、データ送信のためにデータパケットフレームを定義し、データレジスタを設け、制御コマンドによりフロー制御を行いつつ、印刷データをライトトランザクションにより転送する方式である。

【0201】

図58は転送方式1におけるFreeBlockコマンドおよびレスポンスを示す図である。転送方式1においては、データの転送に先立ち、画像供給デバイスが幾つのデータパケットを転送するかを示す情報を取得するためにFreeBlockコマンドおよびレスポンスが用いられる。

【0202】

画像供給デバイスはFreeBlockコマンドをライトトランザクションで転送し(S605-1)、プリンタはそのトランザクションに対する確認を示すACKパケットを返す(S605-2)。プリンタは現在使用可能なFreeBlockCountを通知するためにFree Blockレスポンスを返し(S605-3)、画像供給デバイスはそのトランザクションに対する確認を示すACKパケットを返す(S605-4)。

【0203】

図59は転送方式1におけるデータ転送フローの一例を示す図である。画像供給デバイスは、DPPのLoginコマンドおよびレスポンスによりプリンタにログインし(S606-1およびS606-2)、前記のフォーマットレジスタグループを用いて、データ転送に使用するフォーマットを設定する(S606-3)。その後、OpenChannelコマンドおよびレスポンスにより論理チャネルを開く(S606-4およびS606-5)。続いてデータ転送が開始されるが、印刷データは図55に示したデータパケット形式で転送される。また、パケットの転送は、図57に符号604-2から604-5で示したプリンタ側のデータバッファ数に等しい数のブロック分の転送を一周期として行われる。

【0204】

転送方式1における印刷データの転送は次のように行われる。画像供給デバイスは、FreeBlockコマンドおよびレスポンスによりプリンタのFreeBlock数を得(S606-6およびS606-7)、そのFreeBlock数と同数のデータパケットを、順次、WriteBlockにより転送する(S606-8)。なお、WriteBlockは図54に示したデータレジスタ601-3からデータレジスタ601-9に印刷データ用のパケットを転送するコマンドである。ただし、WriteBlockコマンドにはレスポンスはなく、データレジスタ601-9にデータパケットを格納したことを確認するためのACKパケットがプリンタから返される(S606-9)。

【0205】

そして、FreeBlock数に等しい数の、WriteBlockコマンドによるパケット転送およびそれに対応するACKパケットの返送の繰返しからなるブロック転送が、画像供給デバイスから一連の印刷データがすべて出力されるまで繰返され、各ブロック転送の間には、FreeBlockコマンドおよびレスポンスによるプリンタのFreeBlock数の取得が行われる。

【0206】

印刷データの転送が終了すると、画像供給デバイスは、CloseChannelコマンドおよびレスポンスにより論理チャネルを閉じ（S606-10およびS606-11）、DPPのLogoutコマンドおよびレスポンスによりプリンタからログアウトする（S606-12およびS606-13）。

【0207】

[転送方式2]

Simplified Response Modelである転送方式2は、FreeBlock数を獲得する方法以外は転送方式1と同じ手順でデータ転送を行う。

【0208】

図60は転送方式2におけるデータ転送フローの一例を示す図である。画像供給デバイスは、DPPのLoginコマンドおよびレスポンスによりプリンタにログインし（S607-1およびS607-2）、前記のフォーマットレジスタグループを用いて、データ転送に使用するフォーマットを設定する（S607-3）。その後、OpenChannelコマンドおよびレスポンスにより論理チャネルを開く（S607-4およびS607-5）。続いてデータ転送が開始されるが、印刷データは図55に示したデータパケット形式で転送される。また、パケットの転送は、図57に符号604-2から604-5で示したプリンタ側のデータバッファ数に等しい数のブロック分の転送を一周期として行われる。

【0209】

転送方式2における印刷データの転送は次のように行われる。画像供給デバイスは、WriteBlocksコマンドおよびレスポンスによりプリンタのFreeBlock数を得る（S607-6およびS607-7）。ただし、ステップS607-7のレスポンスは、これ以降

のFreeBlock数の獲得をプリンタ側からのレスポンスのみにより行うために、INT ERIM（暫定）タイプである。画像供給デバイスは、得られたFreeBlock数と同数のデータパケットを、順次、WriteBlockにより転送し（S607-8）、プリンタからは前述したACKパケットが返される（S607-9）。そして、FreeBlock数に等しい数の、WriteBlockコマンドによるパケット転送およびそれに対応するACKパケットの返送の繰返しからなるブロック転送が、画像供給デバイスから一連の印刷データがすべて出力されるまで繰返される。ただし、二周目以降のブロック転送におけるFreeBlock数は、ブロック転送周期の終りごとに、プリンタから送られてくるWriteBlocksレスポンスにより、画像供給デバイスに通知される（S607-10）。このWriteBlocksレスポンスは、プリンタのレスポンスのみによるFreeBlock数の取得を継続するためにCONTINUEタイプである。

【0210】

印刷データの転送が終了すると、画像供給デバイスは、CloseChannelコマンドおよびレスポンスにより論理チャネルを閉じ（S607-11およびS607-12）、DPPのLogoutコマンドおよびレスポンスによりプリンタからログアウトする（S607-13およびS607-14）。

【0211】

[FreeBlock数の取得方法]

以下では、転送方式1と転送方式2の違いであるFreeBlock数の取得方法について詳細に説明する。

【0212】

図61は転送方式1のステップS606-6およびS606-7におけるFreeBlockコマンドおよびレスポンスを詳細に示す図で、図59では省略したライトトランザクションのACKパケットを含めて示したもので、イニシエータデバイス（画像供給デバイス）、ターゲットデバイス（プリンタ）とも、リンクレイヤおよびトランザクションレイヤの処理速度が比較的低速な場合を示している。

【0213】

ライトトランザクションにより、画像供給デバイスがFreeBlockコマンドをコマンドレジスタに書き込む際（S608-1）、プリンタのリンクレイヤからは前述した

`pending`を示すACKパケットが返る(S608-2)。次に、画像供給デバイスは、無データのFreeBlockコマンドを送り(S608-3)、プリンタから`complete`を示すACKパケット(S608-4)を受取って一つのライトトランザクションが終了する。

【0214】

続いて、プリンタは、FreeBlockレスポンスを返すが、これもステップS608-1のFreeBlockコマンドと同様に、FreeBlock数を含むFreeBlockレスポンスとしてレスポンスレジスタに書き込まれる(S608-5)。画像供給デバイスのリンク層からは、`pending`を示すACKパケットが返され(S608-6)、無データのFreeBlockレスポンス(S608-7)が送られ、`complete`を示すACKパケットを受取って(S608-8)、一つのライトトランザクションが終了する。

【0215】

一方、転送方式2におけるFreeBlock数の取得方法は、印刷データのブロック転送サイクルの二周期目以降では、プリンタからのFreeBlockレスポンスのみを用いるので、ステップS608-5からS608-8だけの動作でFreeBlock数を取得することができる。

【0216】

FreeBlock数の取得は、ブロック転送の一周期ごとに必要である。従って、転送方式1よりも転送方式2の方がバス上を転送されるパケットの数を減らすことができる。

【0217】

図62は転送方式1および転送方式2のWriteBlockを詳細に示す図である。WriteBlockはレスポンスを必要としないので、WriteBlock(S609-1)、`pending`を示すACKパケット(S609-2)、無データのWriteBlock(S609-3)および`complete`を示すACKパケット(S609-4)という手順になり、コマンドおよびレスポンスを転送する場合に比べて半分の手順でデータを転送することができ、リンクレイヤとトランザクションレイヤの処理が比較的低速の場合でも比較的高速のデータ転送を実行することができる。

【0218】

図63は転送方式1および転送方式2のWriteBlockを詳細に示す図であるが、リン

クレイヤとトランザクションレイヤの処理が充分高速な場合である。この場合は、WriteBlock(S610-1)、completeを示すACKパケット(S610-2)という手順になり、さらに効率的なデータ転送を実行することができる。

【0219】

図64はバスリセットが発生した場合のWriteBlockのエラー処理を説明する図で、あるブロック転送サイクルのn(=0~255)個目のパケット転送時にバスリセットが発生した場合を示している。ライトトランザクションでは、通常のデータパケット転送の失敗はACKパケットにより検知できるが、バスリセット発生時のエラーは検知できない。そこで本実施形態においては次の手順でエラー処理を実行する。つまり、FreeBlock数を取得し(S611-1)、WriteBlockをn回行い(S611-2から～S611-6)、ここでバスリセットが発生すると(S611-7)、画像供給デバイスは、バスリセット発生直前のWriteBlock[n]を再度行い(S611-8)、その後は通常の処理を続ける(S611-9からS611-14)。

【0220】

[転送方式3]

図65はPUSH Large Buffer Modelにおける画像供給デバイスおよびプリンタのコマンドレジスタ、レスポンスレジスタおよびデータレジスタの構成例を示す図である。

【0221】

FCPに準拠した画像供給デバイスおよびプリンタ間のコマンドおよびレスポンスは、画像供給デバイスのコマンドレジスタ65-1からプリンタのコマンドレジスタ65-4にコマンドの要求データとしてのコマンドフレーム65-7を書込む動作と、プリンタのレスポンスレジスタ65-5から画像供給デバイスのレスポンスレジスタ65-2にレスポンスデータとしてレスポンスフレーム65-8を書込む動作により実行される。

【0222】

また、データフレーム65-9についてはFCPとは異なり、ライトトランザクションを用いて画像供給デバイスのデータレジスタ65-3からプリンタのデータレジスタ65-6へ画像データを書込む一方向動作に、データフレーム65-9が使用される。

【0223】

図66は画像供給デバイスおよびプリンタ間におけるPUSHバッファモデルの動作フローの一例を示す図である。なお、コマンドフレーム、レスポンスフレームにおける、Login、Logout、OpenChannel、CloseChannel、フォーマット設定についての動作は、前述した転送方式1と同様であるから詳細な説明は省略する。

【0224】

図66において、画像供給デバイスは、BufferConfigコマンドのINQUIRYにより、プリンタのバッファ領域について問い合わせる(S701)。これに対してプリンタは、BufferConfigレスポンスにより、バッファサイズやバッファアドレスを返す(S702)。

【0225】

次に、画像供給デバイスは、BufferConfigコマンドのCONTROLにより、データを書込むプリンタのバッファサイズおよびバッファアドレスを設定する(S703)。これに対してプリンタは、BufferConfigレスポンスにより、設定が完了したこと返す(S704)。

【0226】

続いて、画像供給デバイスは、SetBufferコマンドのNOTIFYにより、データの転送を開始することをプリンタに通知する(S705)。これに対してプリンタは、SetBufferレスポンスのINTERIMにより、取り敢えずデータを受信する準備ができたことを返し(S706)、データの転送を開始させる。そして、プリンタは、SetBufferレスポンスのCONTINUEにより、初めに設定されたバッファ領域に対するデータ転送が完了したことを画像供給デバイスに通知する(S709)。

【0227】

ステップS707のWriteBufferは、画像供給デバイスによるデータフレームの書き込み動作を示すもので、プリンタに設定したバッファアドレスに対してデータを順次書き込む動作を行うものである。

【0228】

ステップS708のWriteTransactionResponseは、データフレームを同期転送したときのレスポンスパケットを示す。前述したように、プリンタのデータ読み込み速

度が充分に速ければ、一つのライトトランザクションのアクノリッジを使って処理を完了することができるが、データの取込みに時間がかかる場合はスプリットトランザクションとして独立したレスポンスが発生することになる。

【0229】

ステップS710は複数のデータフレームの転送処理を示す。つまり、BufferConfigコマンドで設定したバッファサイズに対して連続するライトトランザクションによりデータを転送する。このように連続したライトトランザクションを用いるデータ転送方式を「PUSH型データ転送方式」あるいは略して「PUSH方式」と呼ぶ。

【0230】

図67はデータフレームにおけるデータパケットの構成例を示す図で、プリンタのバッファを直接アドレスしてデータを書込むことができるので、データパケット67-1はヘッダなどを含まない構成にすることができる。

【0231】

図68はプリンタのデータレジスタとバッファの関係例を示す図で、データレジスタ68-1に書込まれたデータは、オフセットDestination_Offsetにより決定されるBufferAddressが指すプリンタのメモリ空間68-2のアドレスに直接書込まれる。オフセット値は、データ長Data_Length分ずつインクリメントされるので、連続したバッファアドレスに対して繰返しデータを書込むことで、バッファサイズBufferSizeで示される領域内にデータが連続して書込むことができる。

【0232】

[転送方式4]

図69はPULL Buffer Modelにおける画像供給デバイスおよびプリンタのコマンドレジスタ、レスポンスレジスタおよびデータレジスタの構成例を示す図である。

【0233】

FCPに準拠した画像供給デバイスおよびプリンタ間のコマンドおよびレスポンスは、画像供給デバイスのコマンドレジスタ69-1からプリンタのコマンドレジスタ69-4にコマンドの要求データとしてのコマンドフレーム69-7を書込む動作と、

プリンタのレスポンスレジスタ69-5から画像供給デバイスのレスポンスレジスタ69-2にレスポンスデータとしてレスポンスフレーム69-8を書込む動作により実行される。

【0234】

また、データフレーム69-9についてはFCPとは異なり、リードトランザクションを用いて画像供給デバイスのデータレジスタ69-3からプリンタのデータレジスタ69-6へ画像データを読取る一方向動作に、データフレーム69-9が使用される。

【0235】

図70は画像供給デバイスおよびプリンタ間におけるPULLバッファモデルの動作フローの一例を示す図である。なお、コマンドフレーム、レスポンスフレームにおける、Login、Logout、OpenChannel、CloseChannel、フォーマット設定についての動作は前述した転送方式1と同様であり、BufferConfig、SetBufferのコマンドおよびレスポンス(S711からS714)については前述した転送方式3と同様であるから詳細な説明は省略する。

【0236】

図70において、画像供給デバイスは、SetBufferコマンドのNOTIFYにより、データの転送を開始できることをプリンタに通知する(S715)。これに対してプリンタは、SetBufferレスポンスのINTERIMにより、取り敢えず受信する準備ができたことを返し(S716)、データの転送を開始させる。そして、プリンタは、SetBufferレスポンスのCONTINUEにより、初めに設定されたバッファ領域に対するデータ転送が完了したことを画像供給デバイスに通知する(S719)。

【0237】

プリンタは、PullBufferリクエストにより、リードトランザクションを要求する(S717)。これに対して画像供給デバイスは、PullBufferレスポンスパケットによりデータを転送し(S718)、プリンタに設定されたバッファアドレスにデータが順次書込まれる。

【0238】

ステップS720は複数のデータフレームの転送処理を示す。つまり、BufferConfigコマンドで設定したバッファサイズに対して連続するリードトランザクション

によりデータを転送する。このように連續したリードトランザクションを用いるデータ転送方式を「PULL型データ転送方式」あるいは略して「PULL方式」と呼ぶ。

【0239】

図71は画像供給デバイスのデータレジスタとバッファの関係例を示す図で、データレジスタ71-1に設定されたオフセットDestination_Offsetにより決定されるBufferAddressが指す画像供給デバイスのメモリ空間72-2のアドレスのデータをリードトランザクションにより読み出す。オフセット値は、データ長Data_Length分ずつインクリメントされるので、連續したバッファアドレスに対して繰返しデータを読み出すことで、バッファサイズBufferSizeで示される領域からデータを連続して読み出すことができる。

【0240】

[転送方式5]

Isochronous Modelである転送方式5は、前述した転送方式1の非同期ライトトランザクションを用いた印刷データの転送を、同期ライトトランザクションを用いた印刷データの転送に置き換えたものである。なお、データパケットの構成は、図55および図56に示した構成と同じである。また、ブロック転送におけるプリンタ内のデータパケットフレーム処理は、図57に示した処理と同じである。

【0241】

なお、本転送方式によれば、同期ライトトランザクションを用いることにより、時間を定めてデータ転送することができる。

【0242】

また、ブロック転送することにより、例えば一頁分の印刷データをまとめて転送した際にエラーが発生すると、一頁分の印刷データを再送することになり時間がかかる。しかし、より細かいブロック単位、例えばインクジェットプリンタなどにおける印刷バンド単位で印刷データを転送すればエラー発生にともなう印刷データの再送を効率的に行うことができる。

【0243】

図72はフロー制御のためのコマンドフレームおよびレスポンスフレームを示す

図である。画像供給デバイスは、非同期ライトトランザクションにより、プリンタのコマンドレジスタ75-3にコマンドフレームを書込む。これに対してプリンタは、非同期ライトトランザクションにより、そのコマンドに対するレスポンスマップを画像供給デバイスのレスポンスレジスタ75-2に書込む。

【0244】

図73は印刷処理のフロー例を示す図である。まず、前述した転送方式1と同様に、画像供給デバイスはLoginコマンドをプリンタへ送る(S507)。これに対してプリンタはLoginレスポンスを返し(S508)、コネクションが確立される。

【0245】

次に、前述した転送方式1と同様に、画像供給デバイスは、フォーマット設定を行い(S509)、OpenChannelコマンドをプリンタへ送る(S510)。これに対してプリンタは、OpenChannelレスポンスを返し(S511)、論理チャネルがオープンされる。

【0246】

続いて、画像供給デバイスは、FreeBlockコマンドをプリンタに送る(S512)。これに対してプリンタは、FreeBlockレスポンスを返す(S513)。このFreeBlockレスポンスにはFreeBlock数とErrorStatusが含まれる。FreeBlock数は、プリンタのメモリ空間内にブロック単位で確保されたブロックバッファの数であり、ErrorStatusは前回のブロック転送におけるエラー情報を画像供給デバイスに知らせるためのものである。なお、プリンタは、論理チャネルがオープンされた後の最初のFreeBlockコマンドに対するErrorStatusには常に正常を返す。

【0247】

続いて、画像供給デバイスは、同期トランザクションにより印刷データをブロック転送する(S514)。このとき、画像供給デバイスは、FreeBlock数により示される数分のデータパケットを送出する。

【0248】

次に、画像供給デバイスは、FreeBlockコマンドをプリンタに送る(S515)。これに対してプリンタは、FreeBlockレスポンスを返す(S516)。このレスポンスのErrorStatusが異常を示す場合、つまり先のブロック転送でエラーが生じた場合、

画像供給デバイスはステップS514で転送したデータを再送する(S517)。この後、データ転送が正常に終了するまで、ステップS515、S516およびS517の処理を繰返し実行する。また、ErrorStatusが正常を示す場合、画像供給デバイスはFreeBlockレスポンスに含まれるFreeBlock数のデータパケットを送出する(S517)。

【0249】

なお、プリンタは、転送されるデータのヘッダ部のブロックカウント(図73)を参照することにより、エラーが生じたか否かを認識することができる。また、転送すべきデータのブロック数よりFreeBlock数の方が大きい場合、画像供給デバイスは、その大きい分に相当する数のダミーパケットを送出する。

【0250】

以降、画像供給デバイスから一連の印刷データがすべて出力されるまで同期トランザクションによるデータの転送が繰返される。

【0251】

データの転送が終わると、前述した転送方式1と同様に、画像供給デバイスは、CloseChannelコマンドおよびレスポンスにより論理チャネルを閉じ(S518およびS519)、DPPのLogoutコマンドおよびレスポンスによりプリンタからログアウトする(S520およびS521)。

【0252】

以上説明したように、本実施形態によれば、1394シリアルバスなどにより画像供給デバイスとプリンタを直結し、画像供給デバイスの画像データを直接プリンタへ送り、その画像データに基づく画像をプリンタに印刷させることができる。

【0253】

また、制御コマンドと印刷データとを分離することができるので、1394シリアルバスなどにおいて効率的なデータ転送方式を提供することができる。

【0254】

また、1394シリアルバスにおいて転送エラーを回復することができるデータ転送方式を提供することができる。

【0255】

また、データ転送用のレジスタ領域に関する空きブロック数を通知することに

より、そのレジスタ領域にデータを書込む際の書き込みの可否の判定を不要とし、その判定に要するオーバヘッドを削除したデータ転送方式を提供することができる。さらに、通知された空きブロック数分のデータをまとめて送信し受信するので、転送効率の良いデータ転送方式を提供することができる。

【0256】

また、複数のデータ転送方式の中から転送先のデバイスに合ったデータ転送方式を選択することができるデータ転送方式を提供することができる。

【0257】

また、ホストデバイスからターゲットデバイスへデータを転送する際に、コマンドとそれに対するレスポンスのやり取りを、データ転送の開始を指示するコマンドとそのコマンドに対するレスポンスだけにして、つまり、データ転送の開始後はコマンドの送信を行わないことで、コマンドの送信による転送効率の低下を回避したデータ転送方式を提供することができる。

【0258】

また、ホストデバイスとターゲットデバイス間でデータ転送を行う際に、PUSH方式またはPULL方式によるデータ転送方式を提供することができる。

【0259】

また、ホストデバイスとターゲットデバイス間でデータ転送を行う際に、同期転送と非同期転送と同じ転送手順で行うことができるデータ転送方式を提供することができる。

【0260】

また、ホストデバイスとターゲットデバイス間でデータ転送を行う際に、同期転送においてあるデータ単位で転送エラーが発生した場合に、そのデータ単位について再転送を行うことができるデータ転送方式を提供することができる。

【0261】

また、ホストデバイスとターゲットデバイス間でデータ転送を行う際に、バスリセットが発生した場合でも正しいデータ転送が行うことができるデータ転送方式を提供することができる。

【0262】

さらに、上記のデータ転送方法を利用するプリンタなどの周辺機器を提供することができる。

【0263】

【変形例】

上述した実施形態においては、FCPに基づいたコマンドおよびコマンドに対するレスポンスを用い、レスポンスに情報をセットしてホストデバイスに通知する方法を用いる例を説明したが、IEEE1394のメモリバスモデルの特徴である、メモリ上にレジスタをマッピングする方法も考えられる。

【0264】

この場合、メモリの特定のアドレスに割り当てられたコマンドレジスタにコマンドデータを書き込むことによりコマンドを実行させる。同様に、メモリの特定のアドレスに割り当てられたレスポンスレジスタのデータを読むことでレスポンスが示される。

【0265】

従って、ターゲットデバイスは、コマンドレジスタにコマンドが書き込まれたことを認識すると、そのコマンドを実行し、その後、レスポンスレジスタにその結果や情報を書き込む。ホストデバイスは、コマンドレジスタにコマンドを書き込んだ後、ターゲットデバイスのレスポンスレジスタを読むことで、コマンドの実行結果や情報を得ることができる。

【0266】

このように、メモリバスモデルでのレジスタを用いて本発明を実現することもできる。

【0267】

また、上述した実施形態においてはIEEE1934に規定されるシリアルインタフェイスを用いてネットワークを構成する例を説明したが、本発明はこれに限定されるものではなく、Universal Serial Bus(USB)と呼ばれるシリアルインタフェイスなど、任意のシリアルインタフェイスを用いて構成されるネットワークにも適用することができる。

【0268】

【他の実施形態】

なお、本発明は、複数の機器（例えばホストコンピュータ、インターフェイス機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0269】

また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0270】

また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0271】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0272】

【発明の効果】

以上説明したように、本発明によれば、1394シリアルバスなどによりホストデバイスとターゲットデバイスを直結し、ホストデバイスから直接ターゲットデバイスへ送られる画像データの処理をターゲットデバイスに行わせるのに適したデータ転送装置、データ転送システムおよびその方法、画像処理装置、並びに、記録媒体を提供することができる。

【0273】

また、制御コマンドとデータの分離が可能で転送効率のよいデータ転送装置、データ転送システムおよびその方法、画像処理装置、並びに、記録媒体を提供することができる。

【0274】

また、同期転送と非同期転送と同じ転送手順で行うデータ転送装置、データ転送システムおよびその方法、画像処理装置、並びに、記録媒体を提供することができる。

【図面の簡単な説明】**【図1】**

本発明を適用するシステムの一般的な構成例を示す図、

【図2】

1394シリアルバスによるネットワークの構成例を示す図、

【図3】

1394シリアルバスの構成例を示す図、

【図4】

1394シリアルバスにおけるアドレス空間の一例を示す図、

【図5】

1394シリアルバス用のケーブルの断面を示す図、

【図6】

1394シリアルバスで採用されている、データ転送方式のDS-Link方式を説明するための図、

【図7】

バスリセット信号の発生から、ノードIDが決定し、データ転送が行えるようになるまでの一連のシーケンス例を示すフローチャート、

【図 8】

バスリセット信号の監視からルートノードの決定までの詳細例を示すフローチャート、

【図 9】

ノードID設定の詳細例を示すフローチャート、

【図 10】

1394シリアルバスのネットワーク動作例を示す図、

【図 11】

1394シリアルバスのCSRアーキテクチャの機能を示す図、

【図 12】

1394シリアルバスに関するレジスタを示す図、

【図 13】

1394シリアルバスのノード資源に関するレジスタを示す図、

【図 14】

1394シリアルバスのコンフィギュレーションROMの最小形式を示す図、

【図 15】

1394シリアルバスのコンフィギュレーションROMの一般形式を示す図、

【図 16】

バス使用権の要求を説明する図、

【図 17】

バス使用の許可を説明する図、

【図 18】

1394シリアルバスにおけるアビトレーションの流れを示すフローチャート、

【図 19】

トランザクションレイヤにおけるCSRアーキテクチャに基づくリード、ライト、ロックの各コマンドの要求・レスポンスプロトコルを示す図、

【図 20】

リンクレイヤにおけるサービスを示す図、

【図21】

非同期転送における時間的な遷移を示す図、

【図22】

非同期転送用パケットのフォーマットを示す図、

【図23】

スプリットトランザクションの動作例を示す図、

【図24】

スプリットトランザクションを行う場合の転送状態の時間的遷移例を示す図、

【図25】

同期転送における時間的な遷移を示す図、

【図26】

同期転送用のパケットフォーマット例を示す図、

【図27】

1394シリアルバスにおける同期転送のパケットフォーマットのフィールドの詳細を示す図、

【図28】

同期転送と非同期転送が混在するときの転送状態の時間的遷移を示す図、

【図29】

1394シリアルバスにおけるAV/Cトランザクションの動作を示す図、

【図30】

FCPパケットフレームを含む非同期転送のパケットフォーマットを示す図、

【図31】

AV/Cコマンドフレームの構造を示す図、

【図32】

AV/Cレスポンスフレームの構造を示す図、

【図33】

1394シリアルバスのインターフェイスをOSIモデルの各層と対比させた図、

【図34】

LOGINプロトコルの基本動作を示す図、

【図35】

1394シリアルバスにおける接続形態の一例を示す図、

【図36】

ログイン動作の流れを示す図、

【図37】

LOGINプロトコルのためにターゲットデバイスであるプリンタが備える1394シリアルバスのCSRを示す図、

【図38】

ホストデバイスにおけるログイン処理を示すフローチャート、

【図39】

ターゲットデバイスであるプリンタのログイン処理を示すフローチャート、

【図40】

LOGINプロトコルを実装していないデバイスを考慮した例を説明する図、

【図41】

図40に示した構成とOSIモデルの各層とを対比させた図、

【図42】

レジスタマップを示す図、

【図43】

画像供給デバイスからプリンタへのフレームの流れを示す図、

【図44】

フォーマットレジスタの構成例を示す図、

【図45】

共通レジスタグループのステータスレジスタstatus registerの詳細な構成例を示す図、

【図46】

共通レジスタグループのレジスタGLOBALに保持される情報の詳細例を示す図、

【図47】

共通レジスタグループのレジスタLOCALに保持される情報の詳細例を示す図、

【図48】

レジスタformat[1]に保持される情報の一例を示す図、

【図49】

レジスタformat[2]に保持される情報の一例を示す図、

【図50】

コマンドおよびコマンドに対するレスポンスの一覧を示す図、

【図51】

DPPがサポートする画像データフォーマット例を示す図、

【図52】

フォーマット設定処理の流れを示す図、

【図53】

データ転送方式の設定手順の一例を示す図、

【図54】

転送方式1および転送方式2に必要なレジスタの1394シリアルバスのアドレス空間におけるレジスタマップを示す図、

【図55】

データパケットフレームの一例を示す図、

【図56】

データヘッダの構成例を示す図、

【図57】

ブロック転送におけるプリンタ内のデータパケットフレーム処理を示す図、

【図58】

転送方式1におけるFreeBlockコマンドおよびレスポンスを示す図、

【図59】

転送方式1におけるデータ転送フローの一例を示す図、

【図60】

転送方式2におけるデータ転送フローの一例を示す図、

【図61】

転送方式1におけるFreeBlockコマンドおよびレスポンスを詳細に示す図、

【図62】

転送方式1および転送方式2のWriteBlockを詳細に示す図、

【図63】

転送方式1および転送方式2のWriteBlockを詳細に示す図、

【図64】

バスリセットが発生した場合のWriteBlockのエラー処理を説明する図、

【図65】

PUSH Large Buffer Modelにおける画像供給デバイスおよびプリンタのコマンドレジスタ、レスポンスレジスタおよびデータレジスタの構成例を示す図、

【図66】

画像供給デバイスおよびプリンタ間におけるPUSHバッファモデルの動作フローの一例を示す図、

【図67】

データフレームにおけるデータパケットの構成例を示す図、

【図68】

プリンタのデータレジスタとバッファの関係例を示す図、

【図69】

PULL Buffer Modelにおける画像供給デバイスおよびプリンタのコマンドレジスタ、レスポンスレジスタおよびデータレジスタの構成例を示す図、

【図70】

画像供給デバイスおよびプリンタ間におけるPULLバッファモデルの動作フローの一例を示す図、

【図71】

画像供給デバイスのデータレジスタとバッファの関係例を示す図、

【図72】

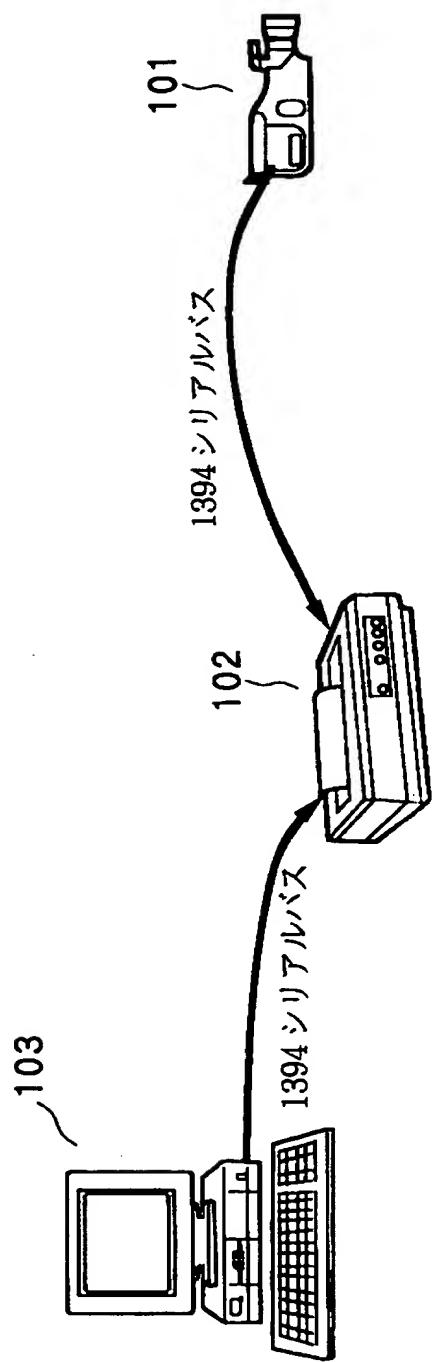
フロー制御のためのコマンドフレームおよびレスポンスフレームを示す図、

【図73】

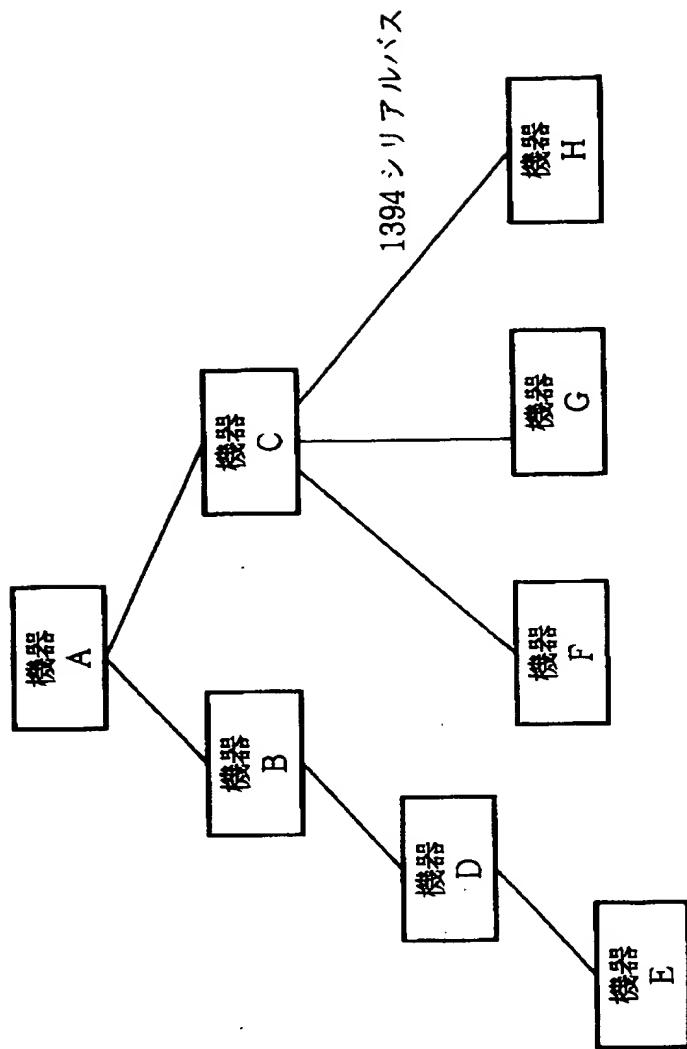
印刷処理のフロー例を示す図である。

【書類名】 図面

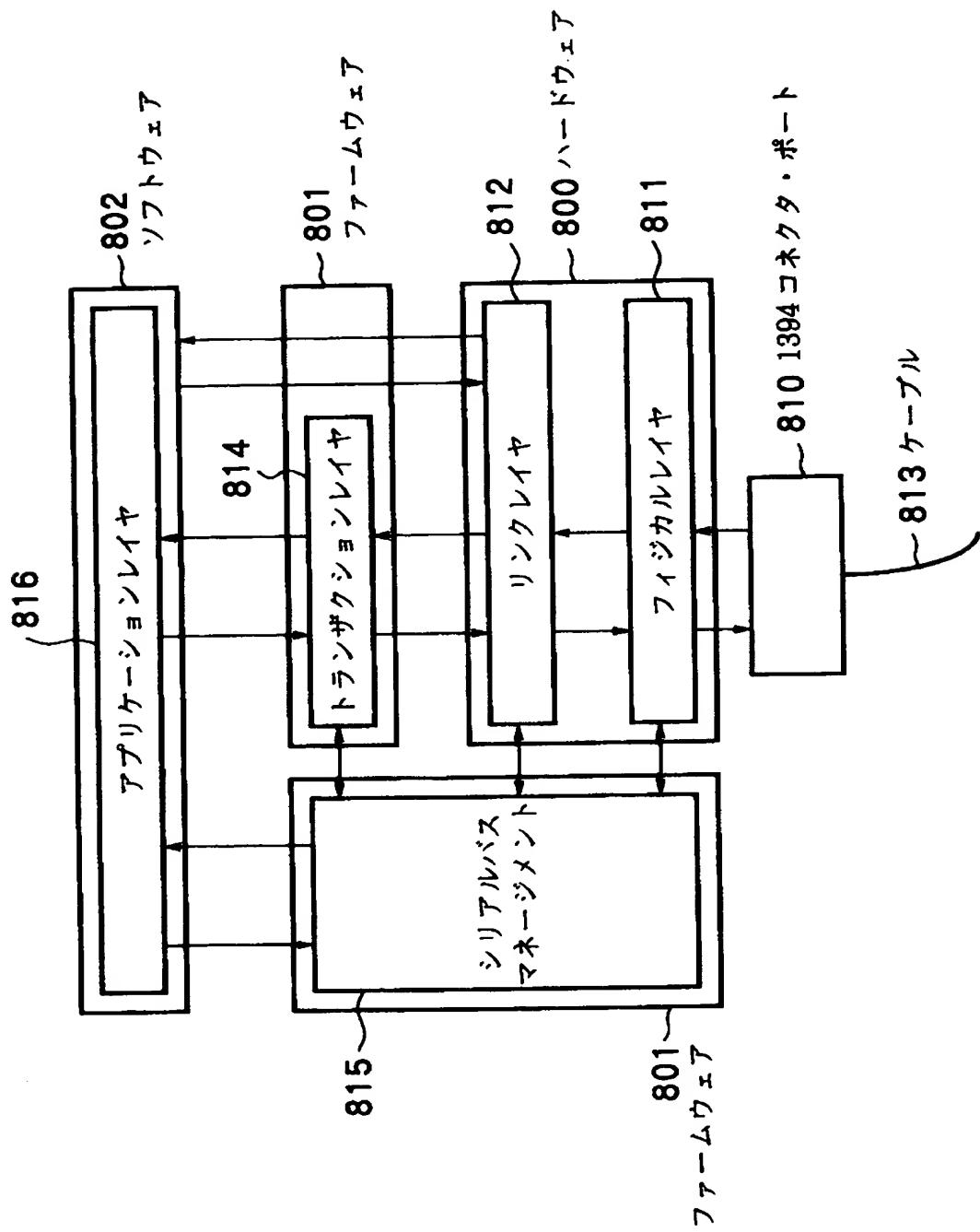
【図 1】



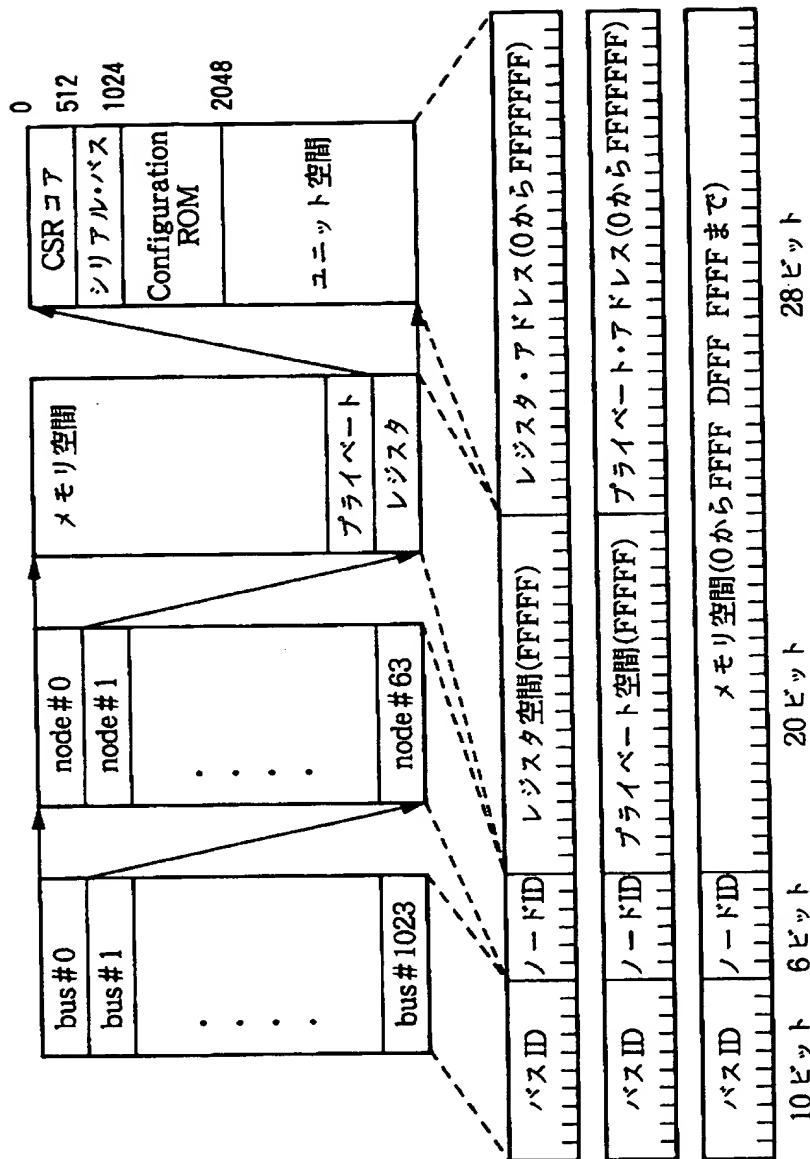
【図2】



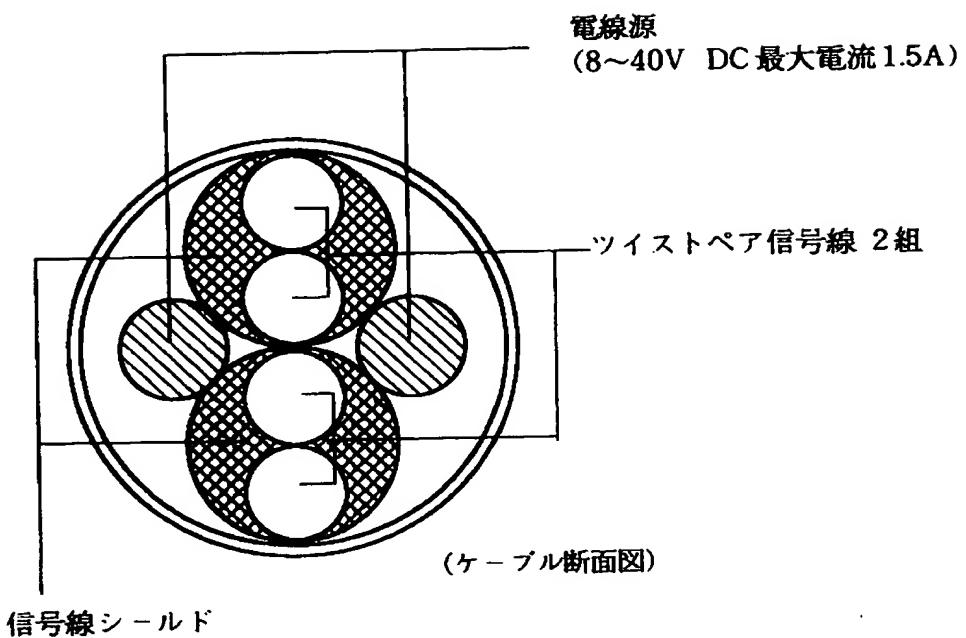
【図3】



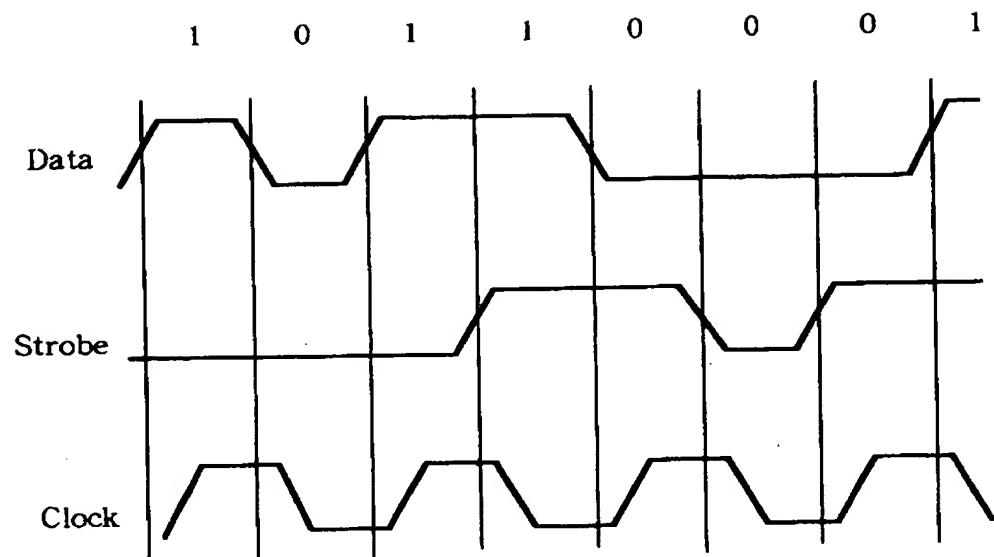
【図4】



【図5】

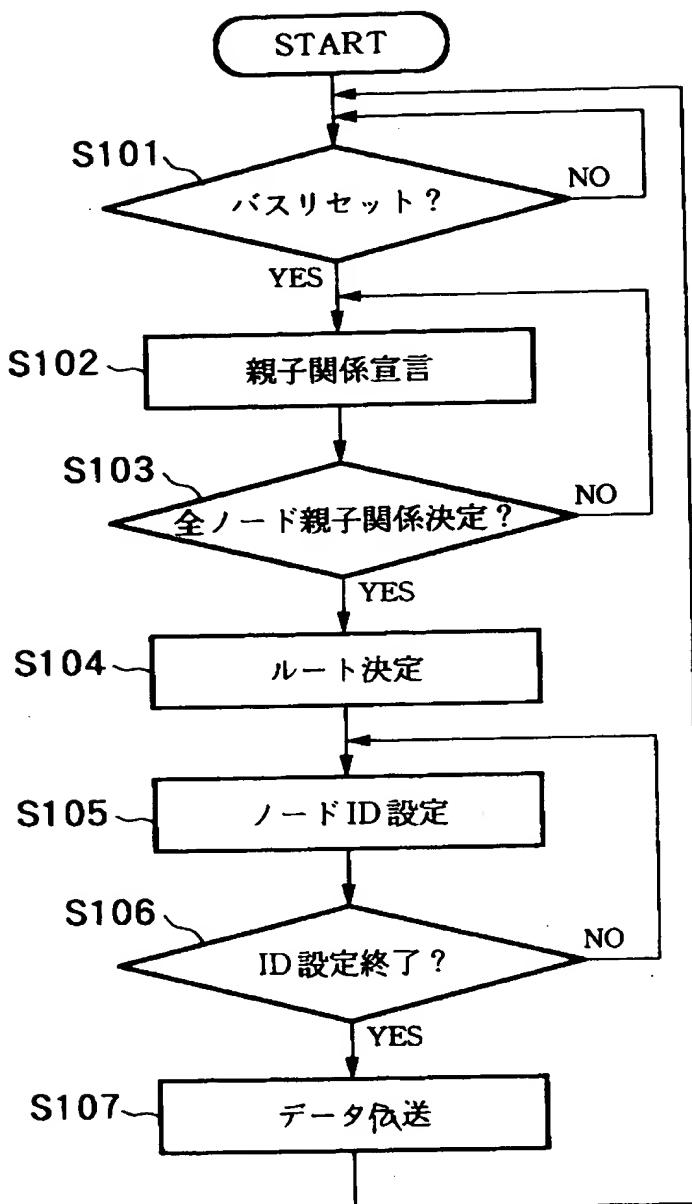


【図6】

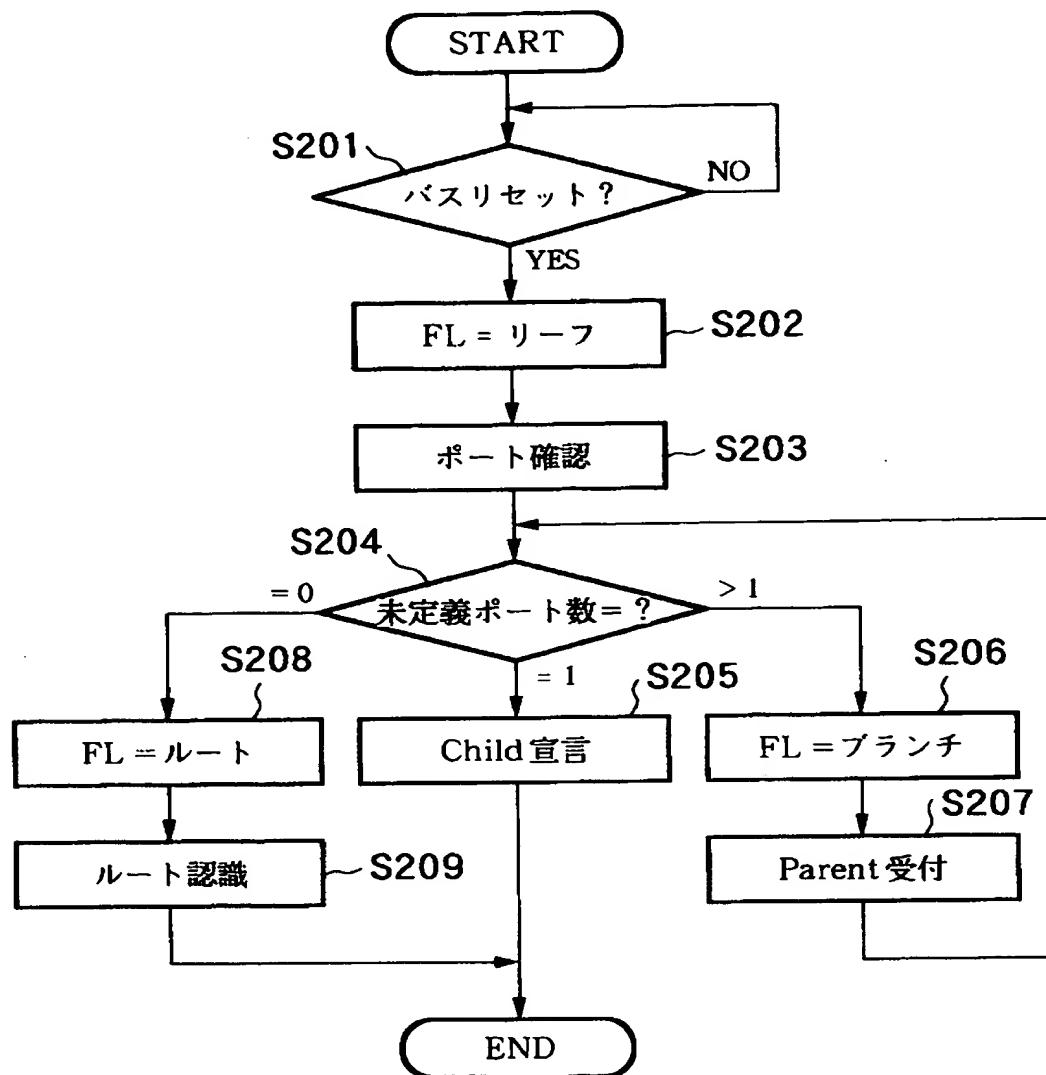


(Data と Strobe の排他的論理和信号)

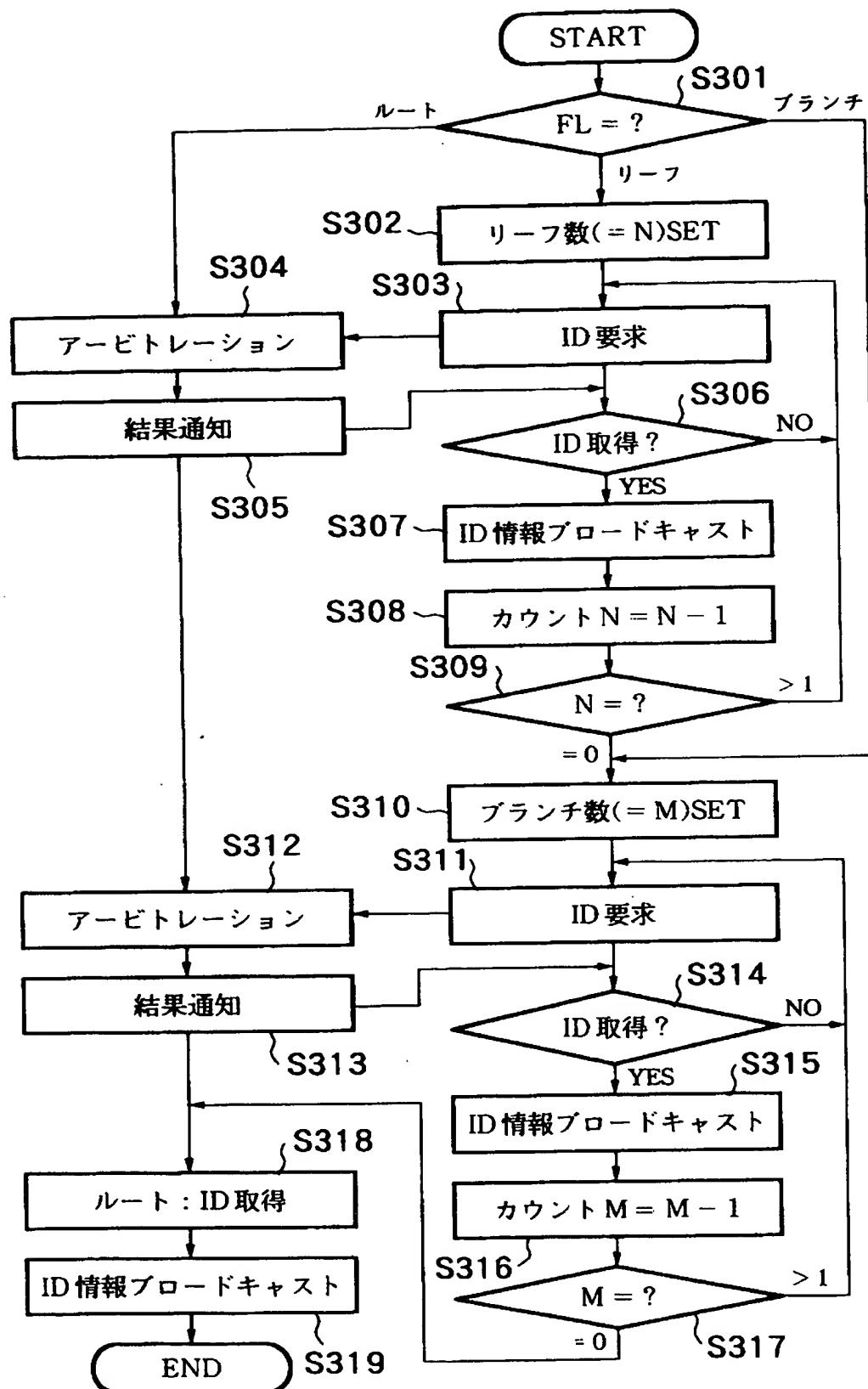
【図7】



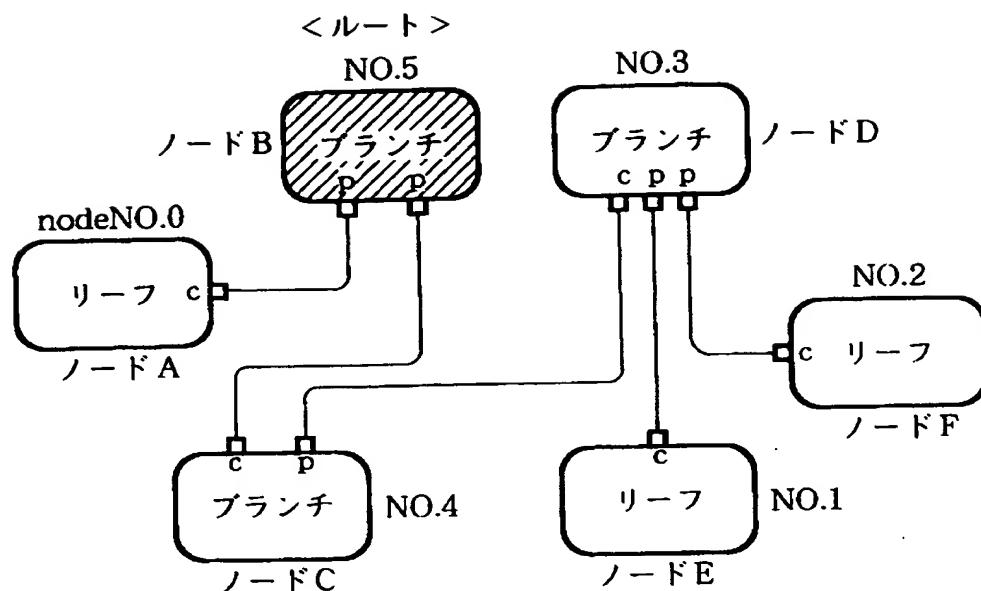
【図8】



【図9】



【図10】



ブランチ：2つ以上のノード接続があるノード
 リーフ：1つのポートのみ接続があるノード

□：ポート
 c：子のノードに相当するポート
 p：親のノードに相当するポート

【図11】

CSRコア・レジスタ

オフセット (16進数)	レジスタ名称	機能
000	STATE_CLEAR	状態と制御の情報
004	STATE_SET	STATE_CLEARの書き込み可否を示す情報
008	NODE_IDS	バスID+ノードID
00C	RESET_START	この領域に対する書き込みでバスをリセット
010~014	INDIRECT_ADDRESS, INDIRECT_DATA	1Kより大きいROMをアクセスするためのレジスタ
018~01C	SPLIT_TIMEOUT	スプリット・トランザクションのタイムアウトを検出するタイマの値
020~02C	ARGUMENT, TEST_START, TEST_STATUS	診断用のレジスタ
030~04C	UNITS_BASE, UNITS_BOUND, MEMORY_BASE, MEMORY_BOUND	IEEE1394では、実装しない
050~054	INTERRUPT_TARGET, INTERRUPT_MASK	割り込み通知レジスター
058~07C	CLOCK_VALUE, CLOCK_TICK_PERIOD, CLOCK_STROBE_ARRIVED, CLOCK_INFO	IEEE1394では、実装しない
080~0FC	MESSAGE_REQUEST, MESSAGE_RESPONSE	メッセージ通知レジスター
100~17C		予約
180~1FC	ERROR_LOG_BUFFER	IEEE1394用に予約

【図12】

シリアル・バス・レジスタ

オフセット (16進数)	レジスタ名	機能
200	CYCLE_TIME	アイソクロナス転送のためのカウント
204	BUS_TIME	時間を同期するためのレジスタ
208	POWER_FAIL_IMMINENT	電源供給に関するレジスタ
20C	POWER_SOURCE	
210	BUSY_TIMEOUT	トランザクション層の再試行を制御
214 ↓ 218		予約
21C	BUS_MANAGER_ID	バス・マネージャのノードID
220	BANDWIDTH_AVAILABLE	アイソクロナス転送の帯域を管理
224 ↓ 228	CHANNELS_AVAILABLE	アイソクロナス転送のチャネル番号を管理
22C	MAINT_CONTROL	
230	MAINT.Utility	診断用レジスタ
234 ↓ 3FC		予約

【図13】

シリアル・バス装置レジスタ

オフセット (16進数)	レジスタ名称	機能
800 ↓ FFC		予約
1000 ↓ 13FC	TOPOLOGY_MAP	シリアル・バスの構成情報
1400 ↓ 1FFC		予約
2000 ↓ 2FFC	SPEED_MAP	シリアル・バスの転送速度の 情報
3000 ↓ FFFC		予約

【図14】

最小形式の Configuration ROM

01	ペンドID
----	-------

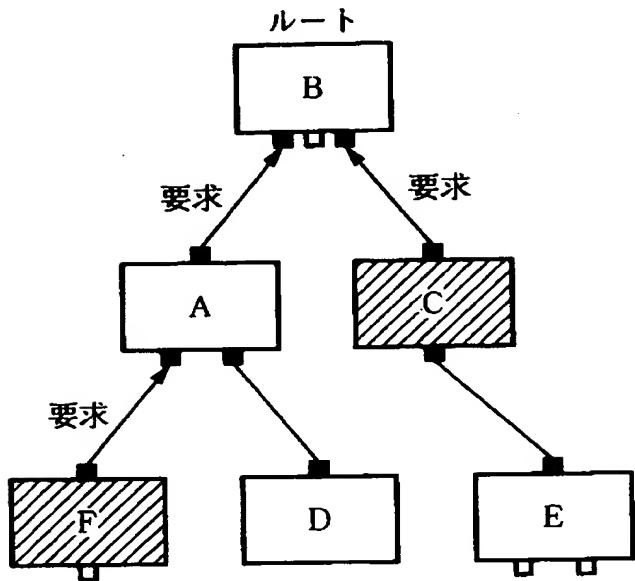
【図15】

一般形式のConfiguration ROM

bus_info_block の長さ	ROM の長さ	CRC
bus_info_block (1394 の ASCII コードと、ノードがアイソクロナス資源管理、サイクル・マスター・バス・マネージャの能力をもつているかの情報)		
root_directory (ベンダIDとノードの機能を表す)		
unit_directories (ユニットの種類とドライバ・ソフトのバージョン)		
root & unit leaves		
vender_dependent_information		

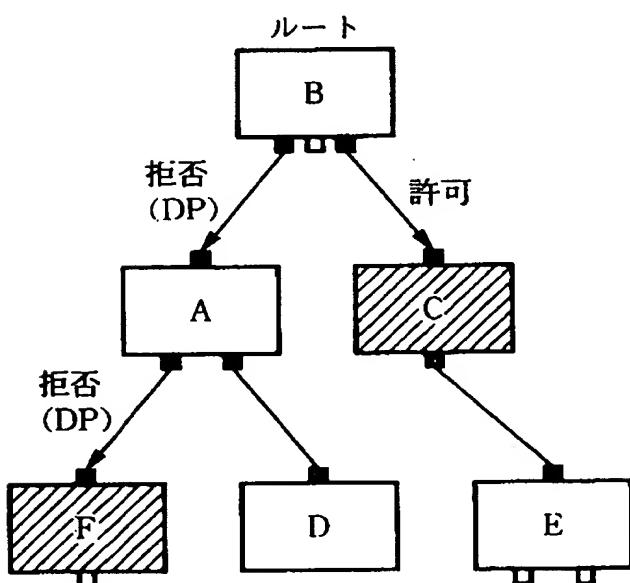
【図16】

バス使用権の要求

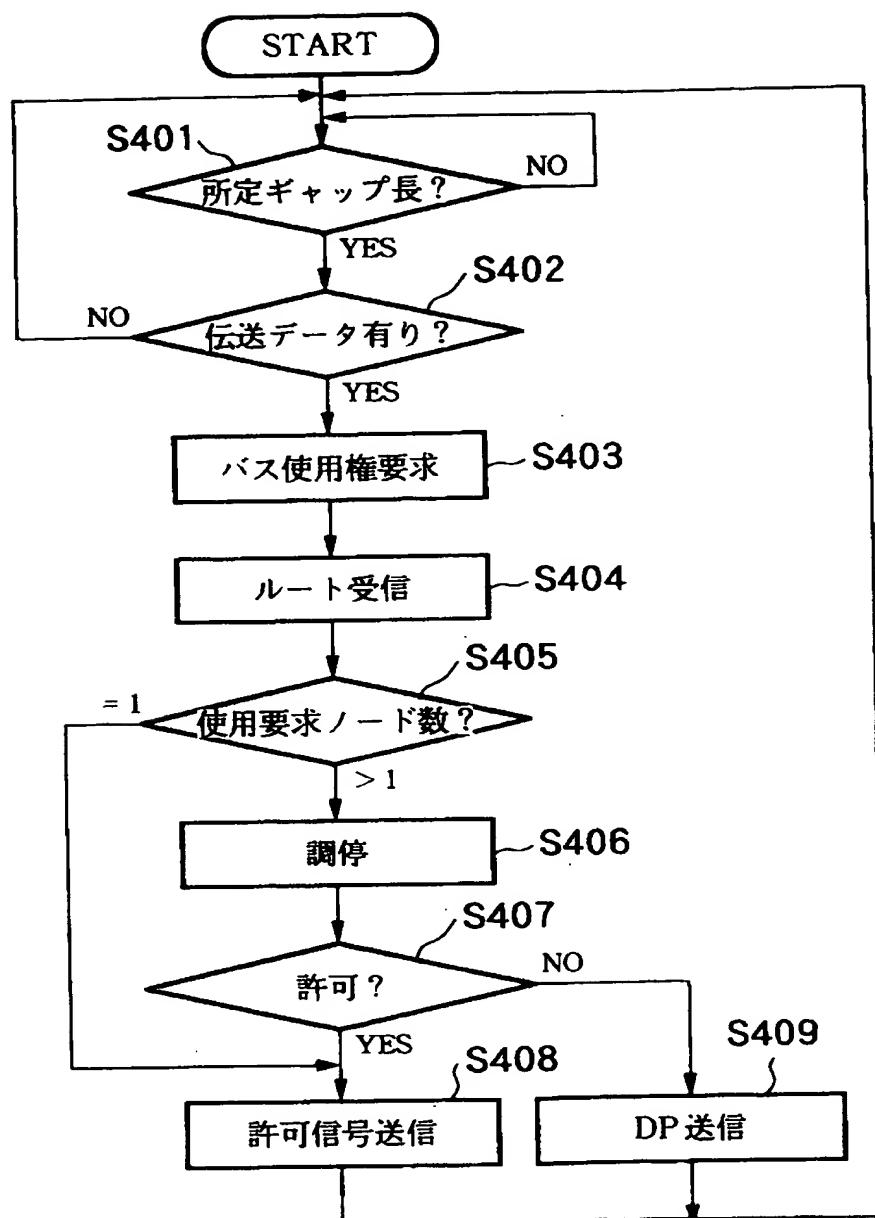


【図17】

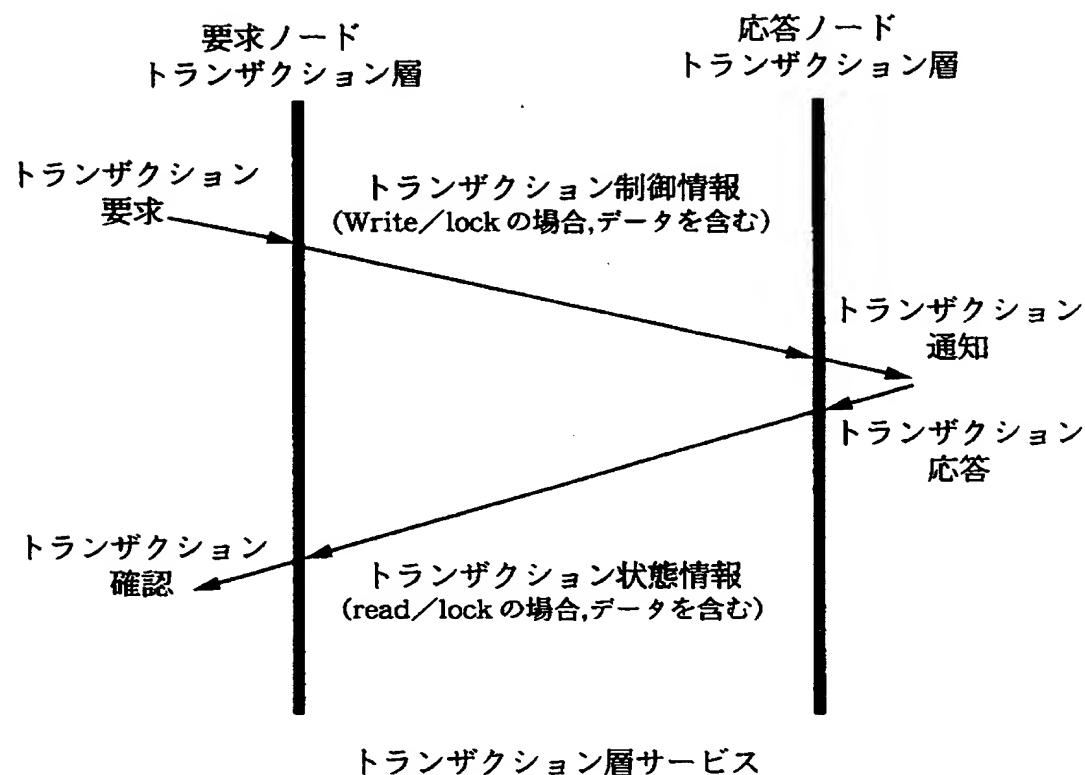
バス使用の許可



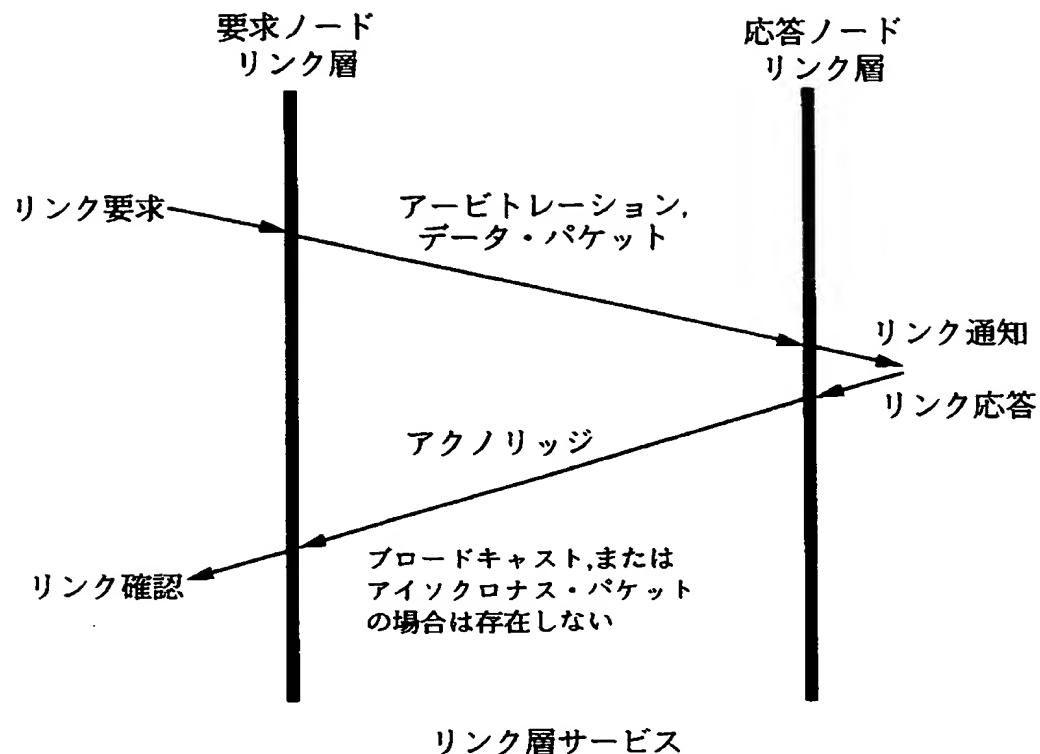
【図18】



【図19】



【図20】



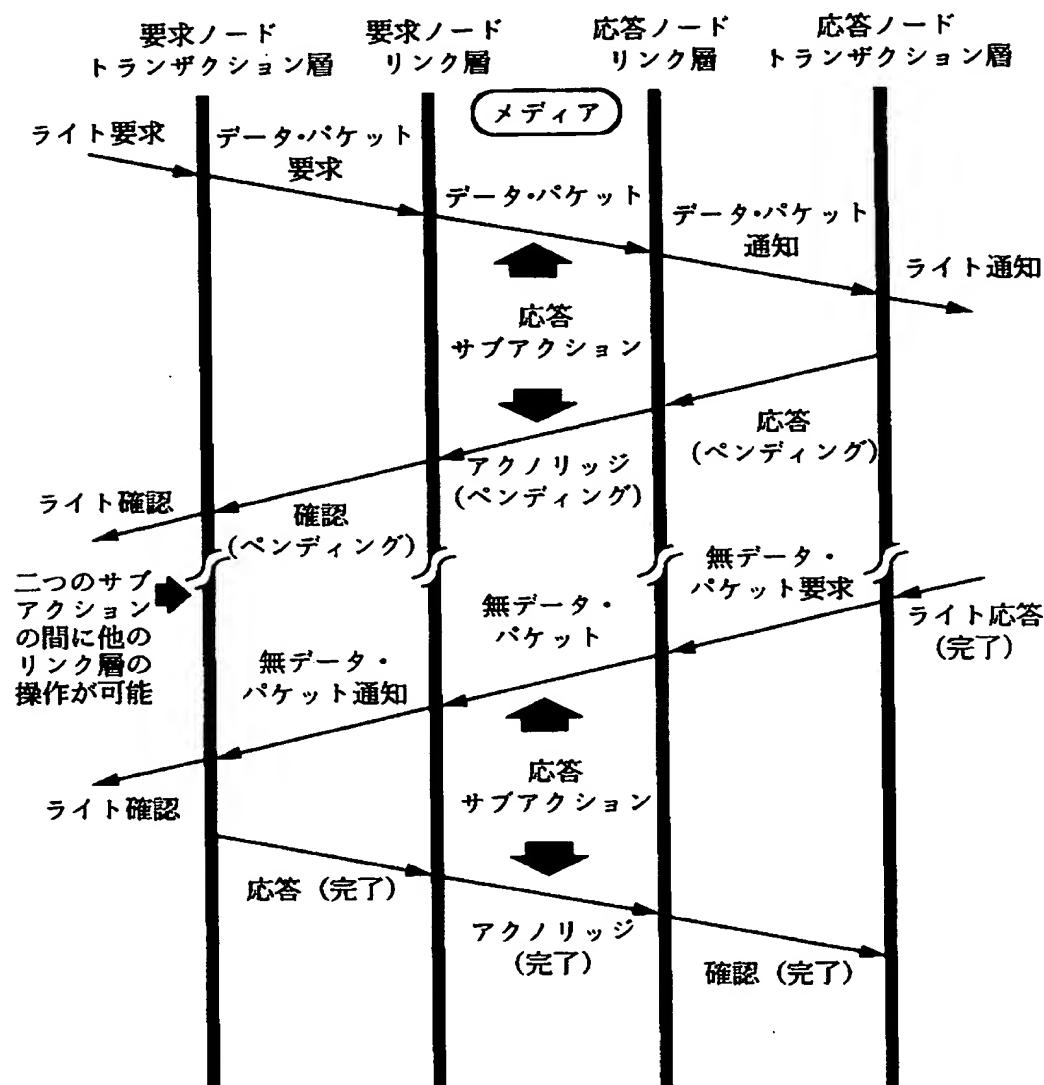
【図21】



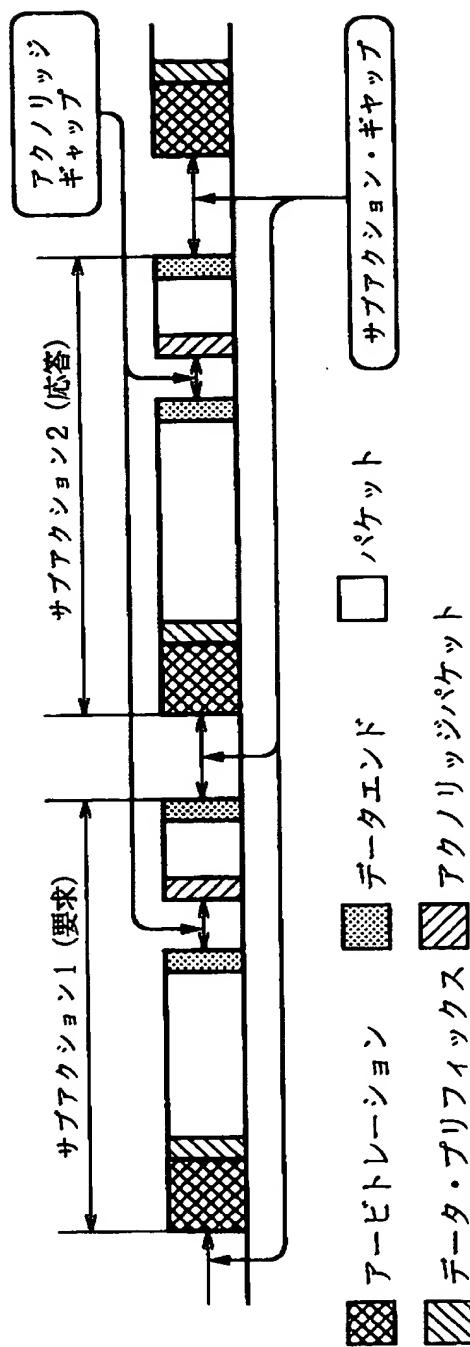
【図22】

destination_ID	t1	rt	tcode	pri
source_ID				
destination_offset				
data_length	extended_tcode			
	header_CRC			
	data_field			
	pad_field			
	data_CRC			

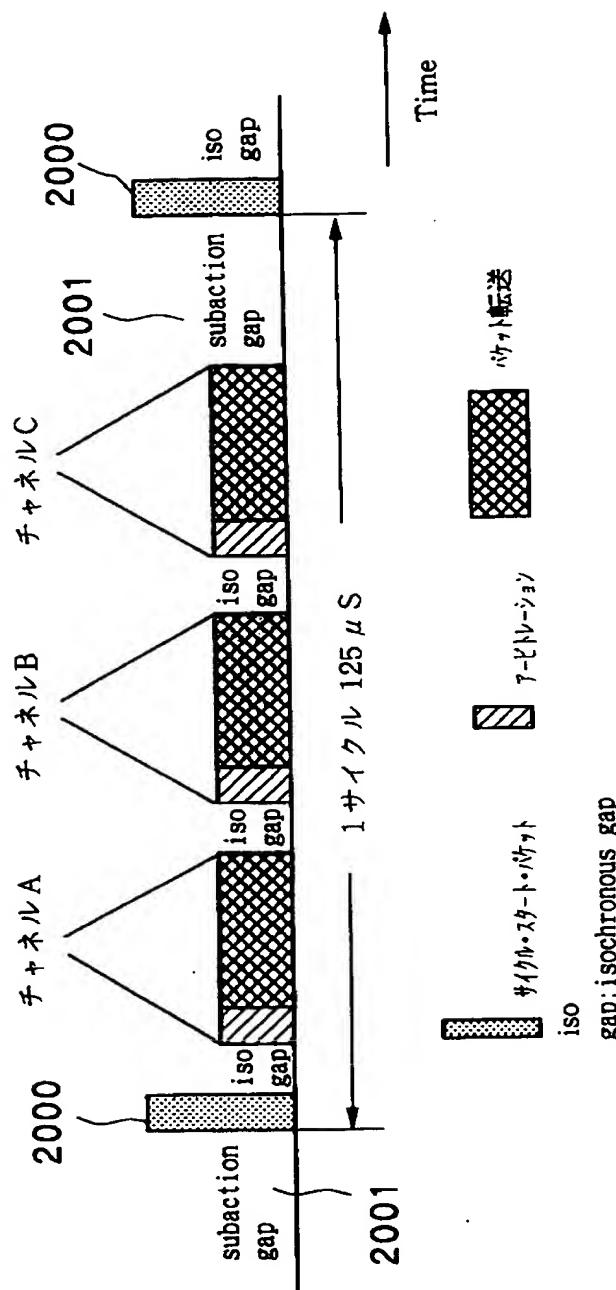
【図23】



【図24】



【図25】



【図26】

アイソクロナス・データのパッケージ

```

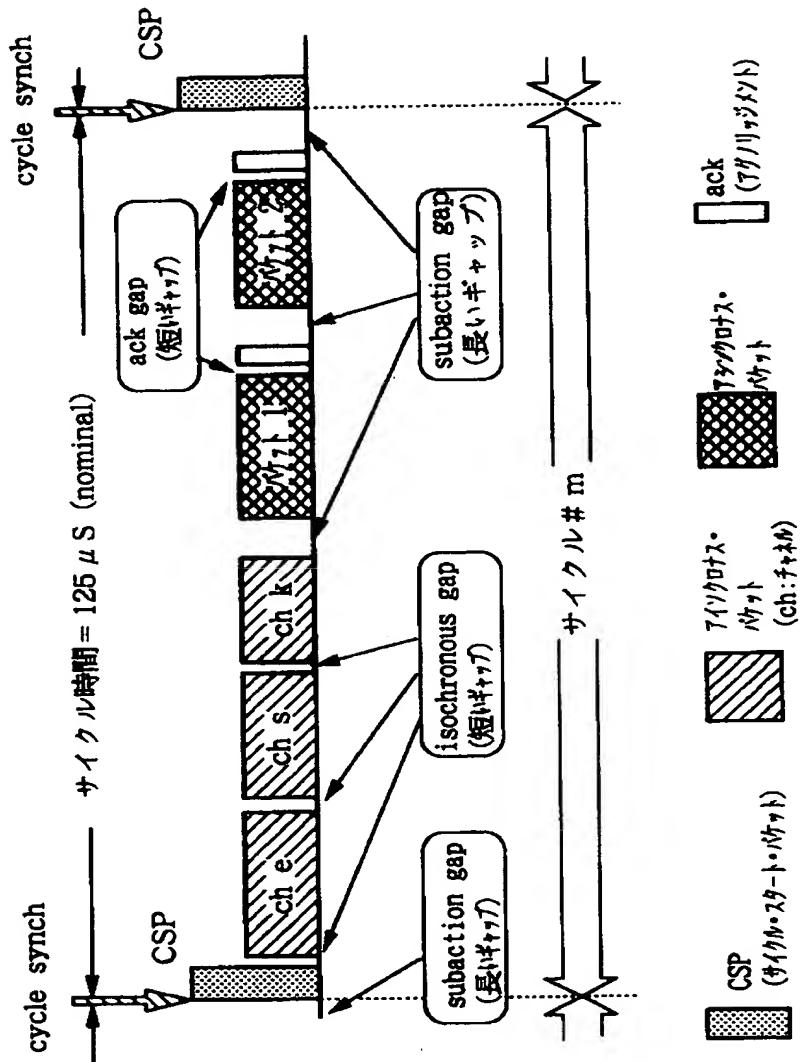
sequenceDiagram
    participant Data as Data
    participant Bytes as Bytes[Raw bytes]
    Bytes->>Data: +data_length, +tag, +channel, +tcode, +sy
    activate Data
    Note over Bytes: header_CRC  
data_field  
pad_field
    deactivate Data
    Data-->>Bytes: data_CRC
  
```

The diagram illustrates the construction of a `Data` object from raw bytes. It begins with a `Bytes` participant sending five initial attributes (`+data_length`, `+tag`, `+channel`, `+tcode`, and `+sy`) to the `Data` participant. A note above the `Bytes` participant specifies three fields: `header_CRC`, `data_field`, and `pad_field`. After receiving these attributes, the `Data` participant sends a return message (`data_CRC`) back to the `Bytes` participant.

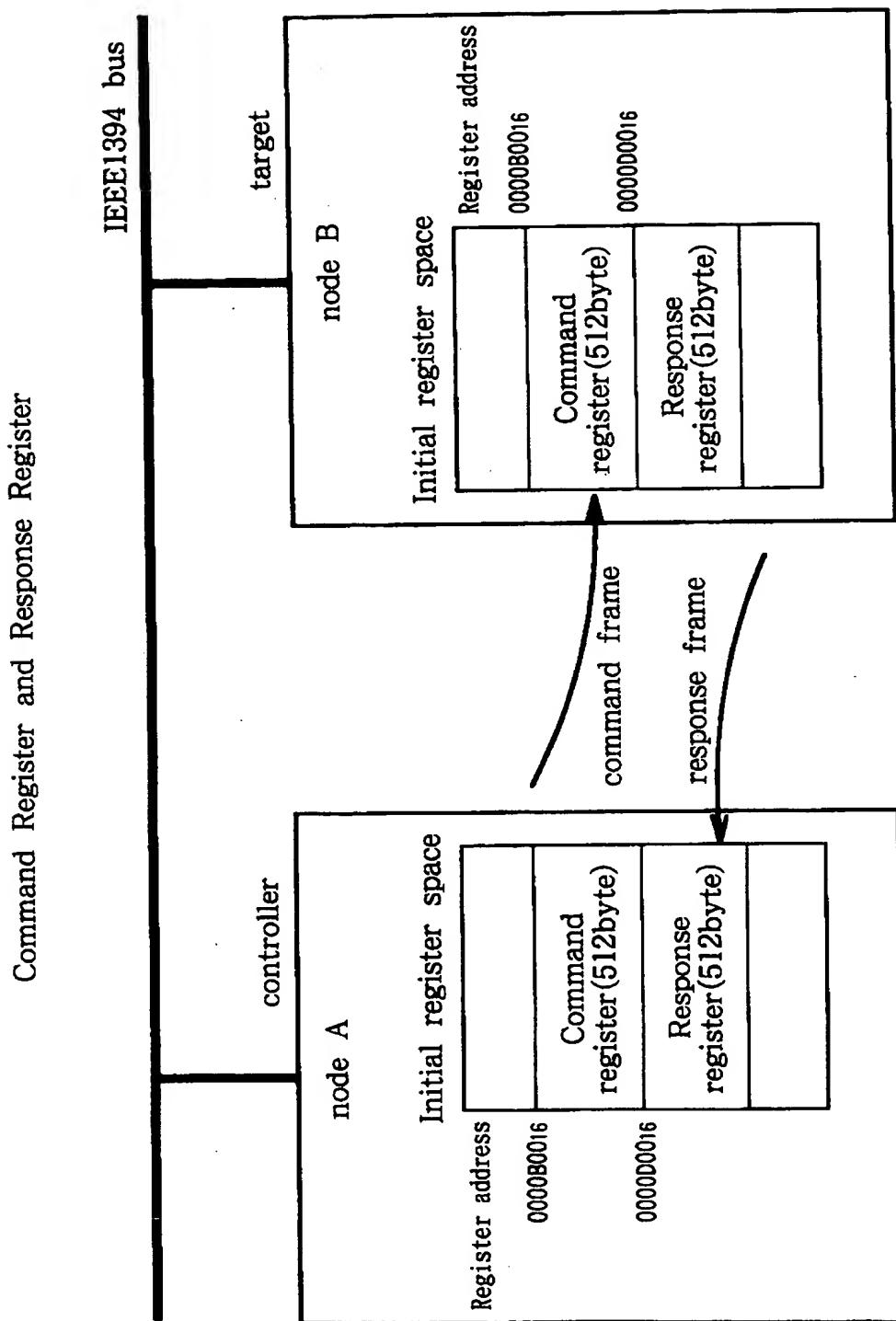
【図27】

略称	名称	内容
destination_ID	destination identifier	着信先ノードのIDを示す (アシンクロナスのみ)
t1	transaction label	一連のトランザクションを示すため のラベル (アシンクロナスのみ)
rt	retry code	再送ステータスを示すコード (アシンクロナスのみ)
tcode	transcation code	パケットの種別を示すコード (アシンクロナスのみ)
pri	priority	優先順位 (アシンクロナスのみ)
source_ID	source identifier	発信元ノード (アシンクロナスのみ)
destination_offset	destination memory address	着信先ノードのメモリ・アドレス (アシンクロナスのみ)
rcode	response code	応答ステータス (アシンクロナスのみ)
quadiet_data	quadiet(4bytes) data	4バイト長のデータ (アシンクロナスのみ)
data_length	length of data	data_fieldの長さ (pad bytesは除く)
extended_tcode	extended transaction code	拡張トランザクション・コード (アシンクロナスのみ)
chanel	isochronous identifier	アイソクロナス・パケットの識別を行なう
sy	synchronization code	映像・音声などの同期に使われる (アシンクロナスのみ)
cycle_time_data	contents of the CYCLE_TIME register	サイクル・マスター・ノードのサイクル・ タイマ・レジスタの値 (サイクル・パケットのみ)
data_field	data + pad bytes	データが格納される (アイソクロナスとアシンクロナス)
header_CRC	CRC for header field	ヘッダ部分に対するCRC
data_CRC	CRC for data field	データ部分に対するCRC
tag	tag label	アイソクロナス・パケットの フォーマット

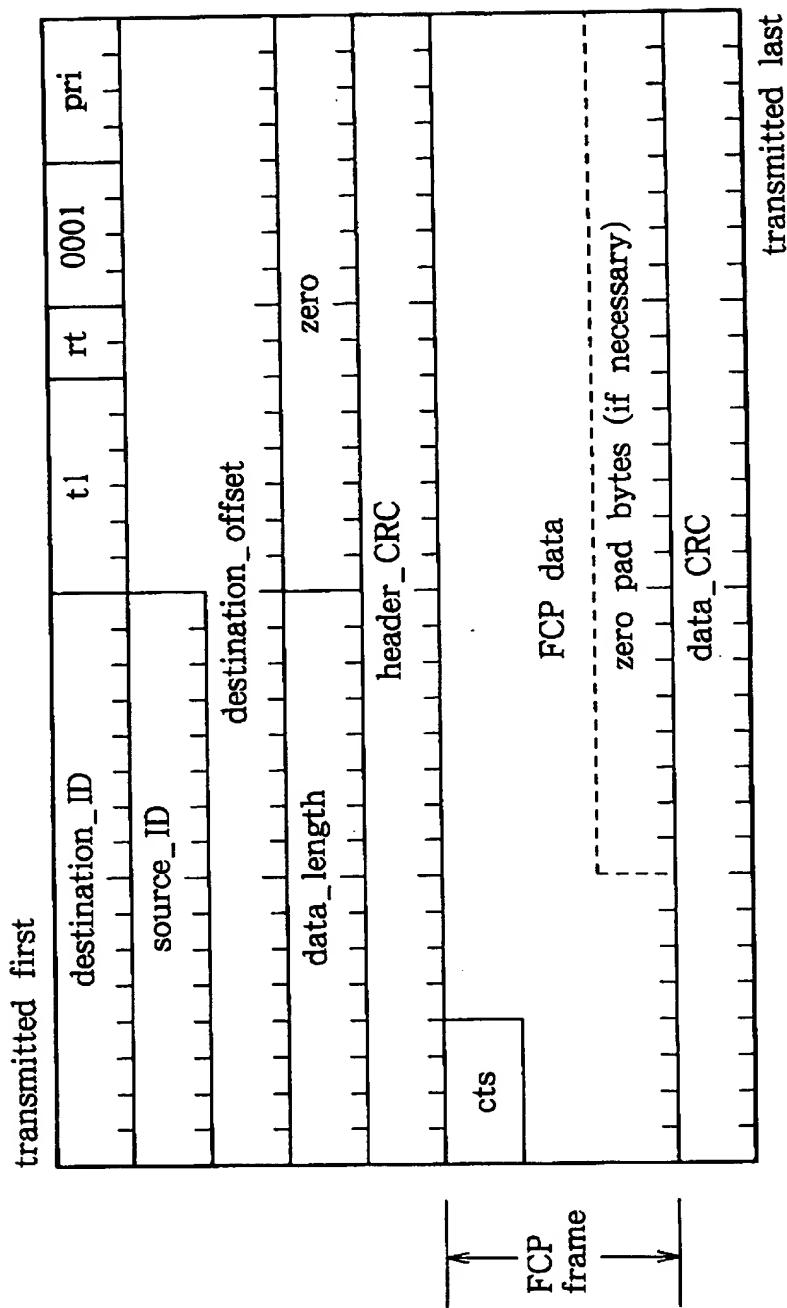
【図28】



【図29】

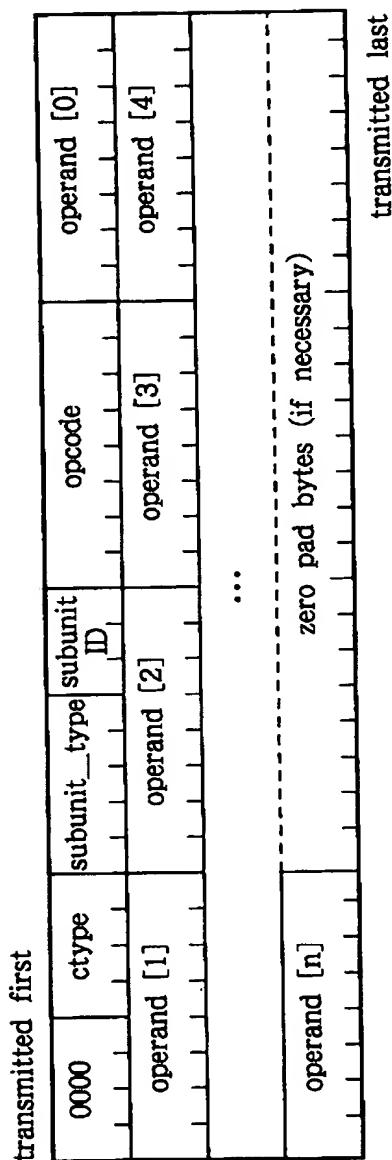


【図30】



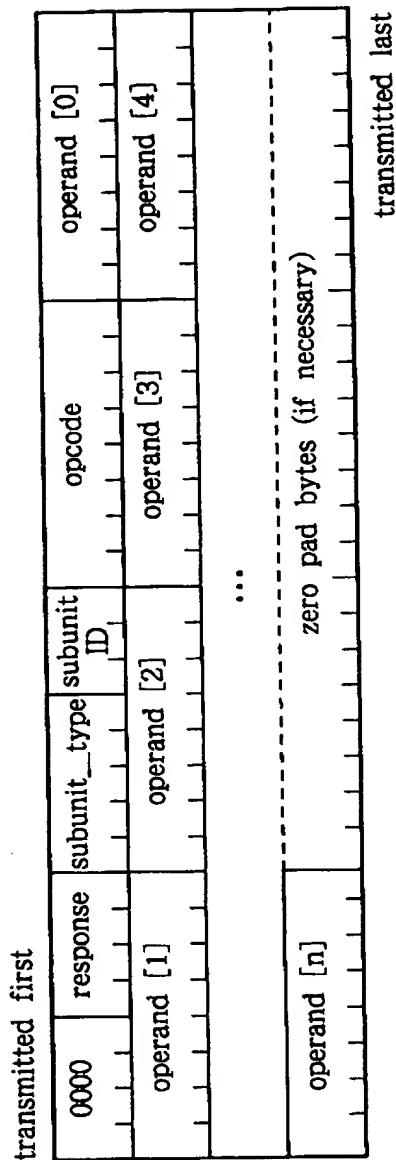
【図31】

AV/C command frame

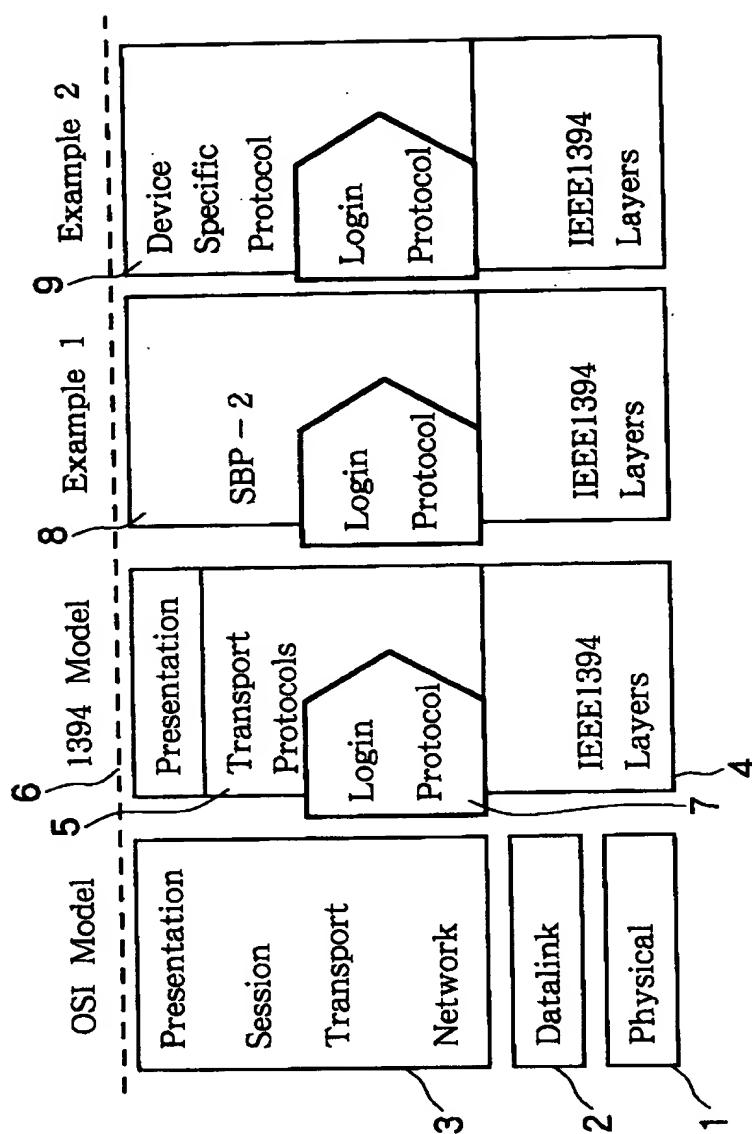


【図32】

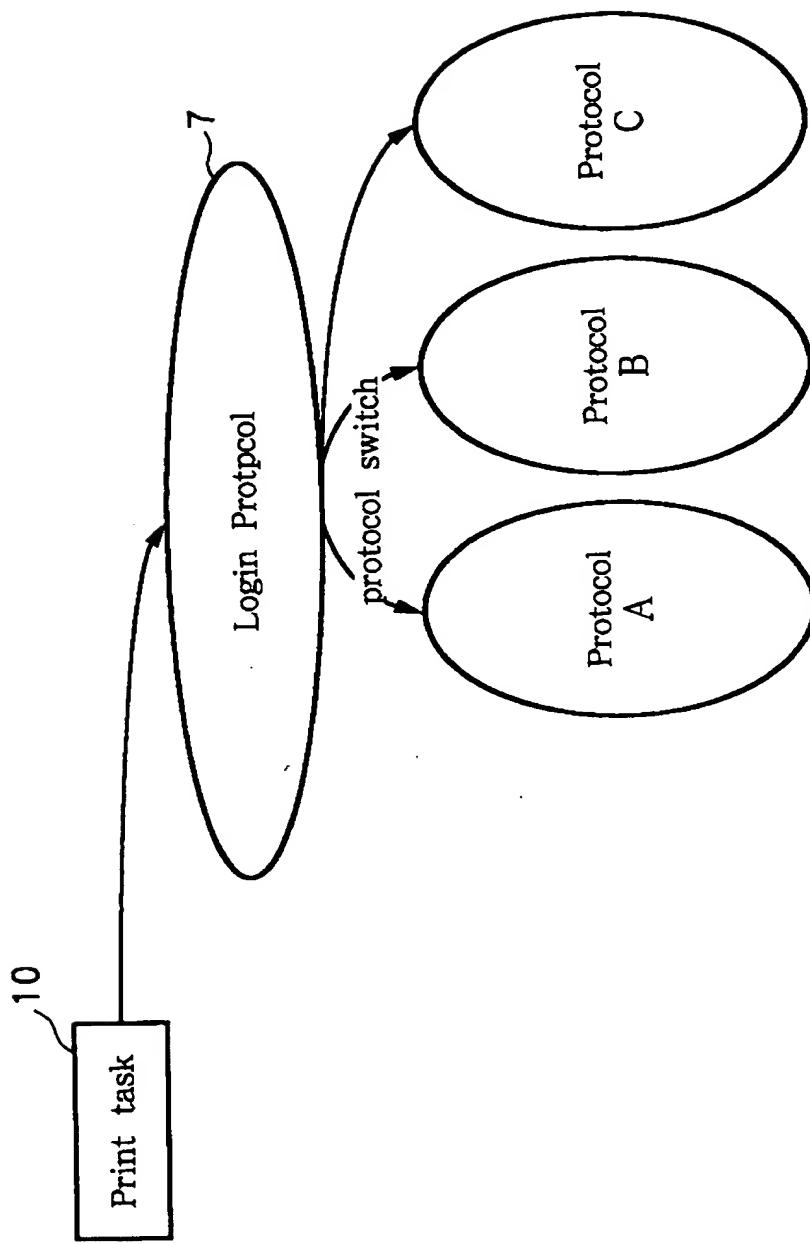
AV/C response frame



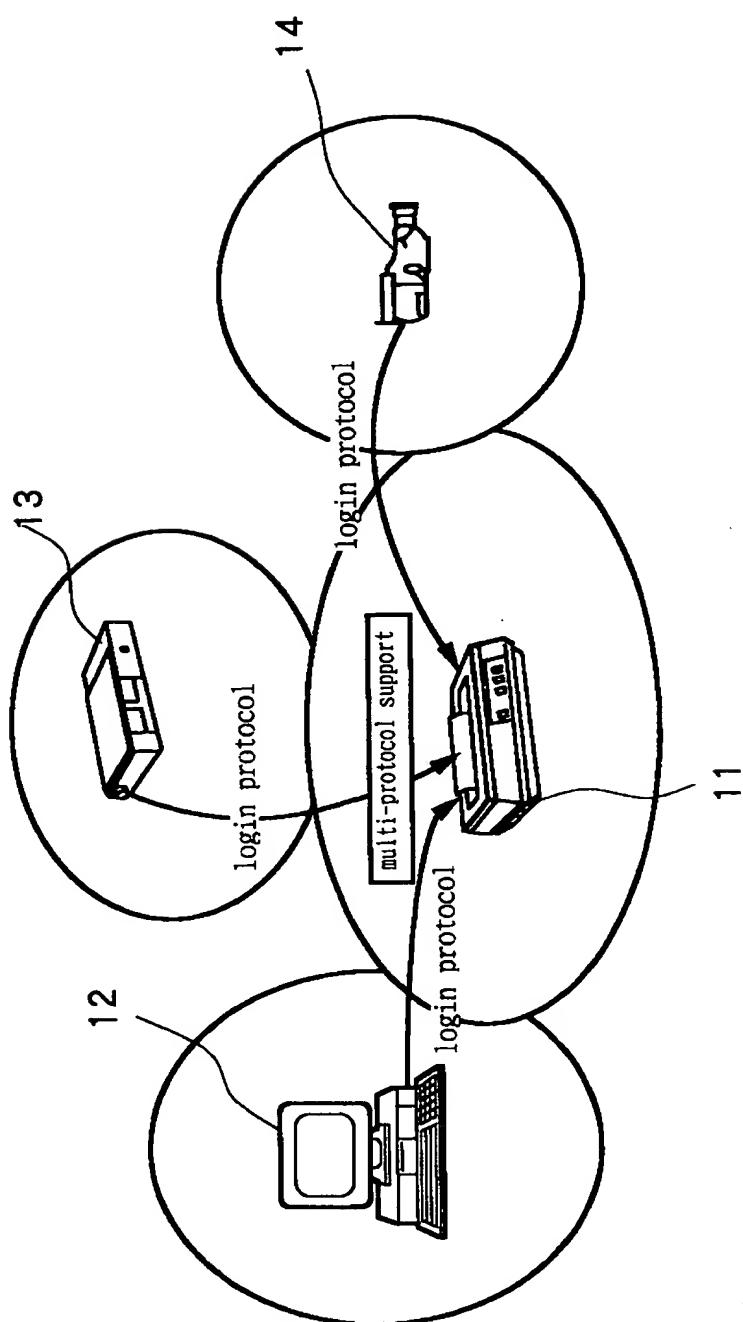
【図33】



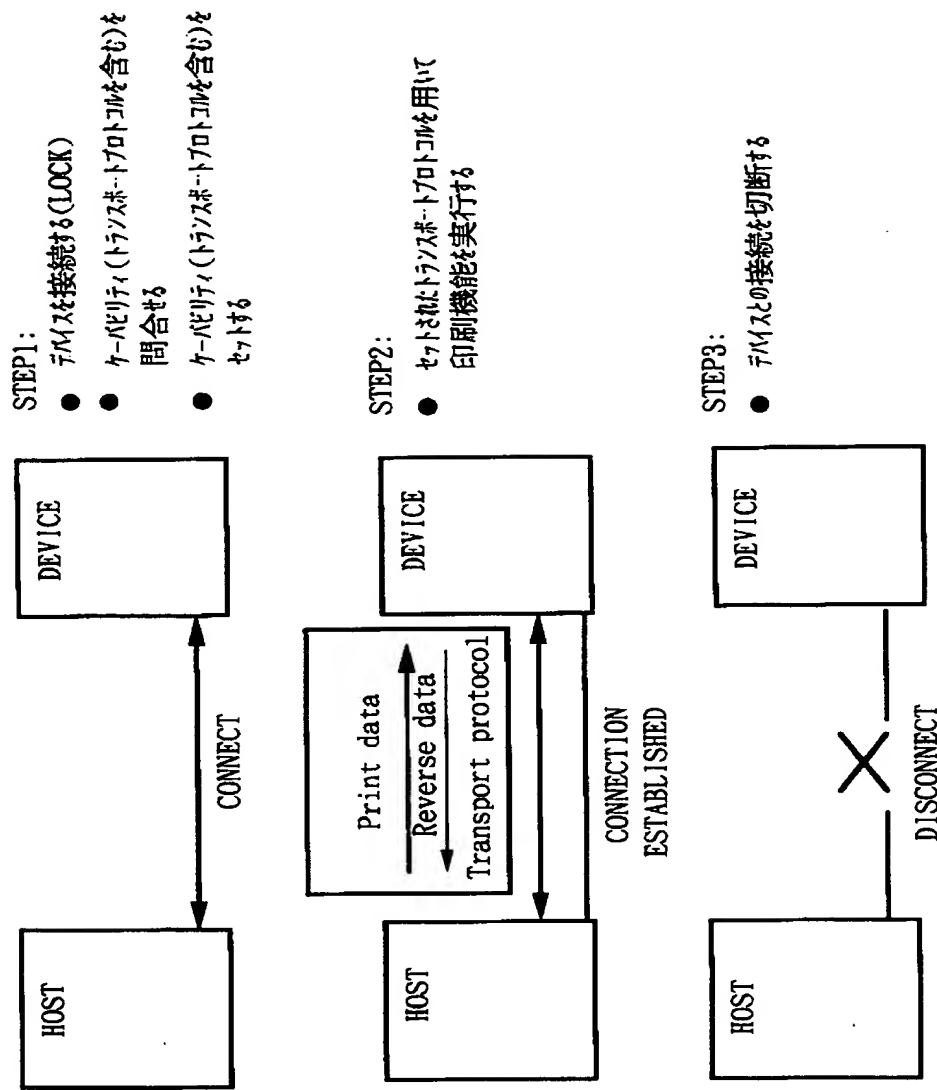
【図34】



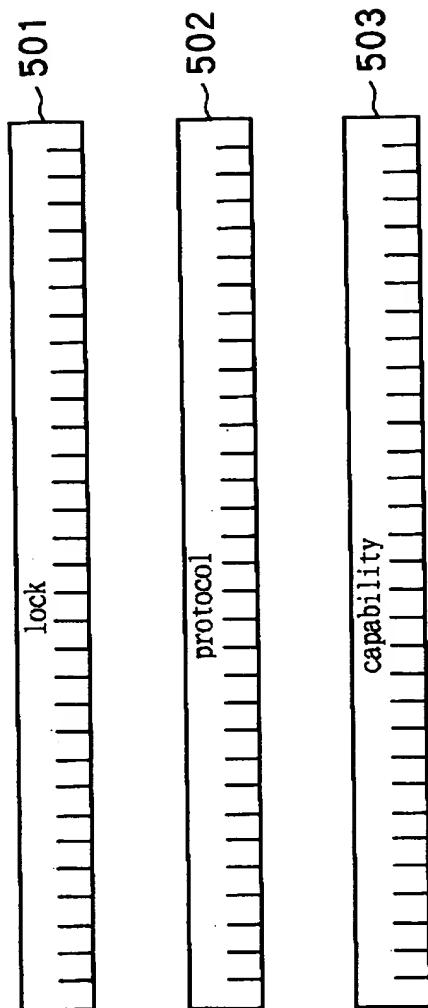
【図35】



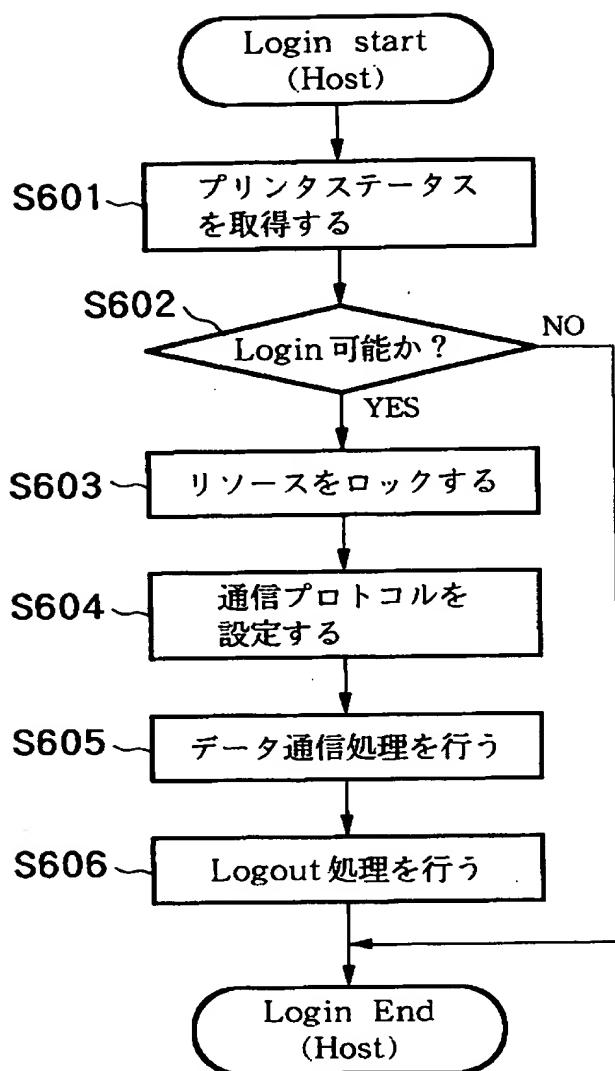
【図36】



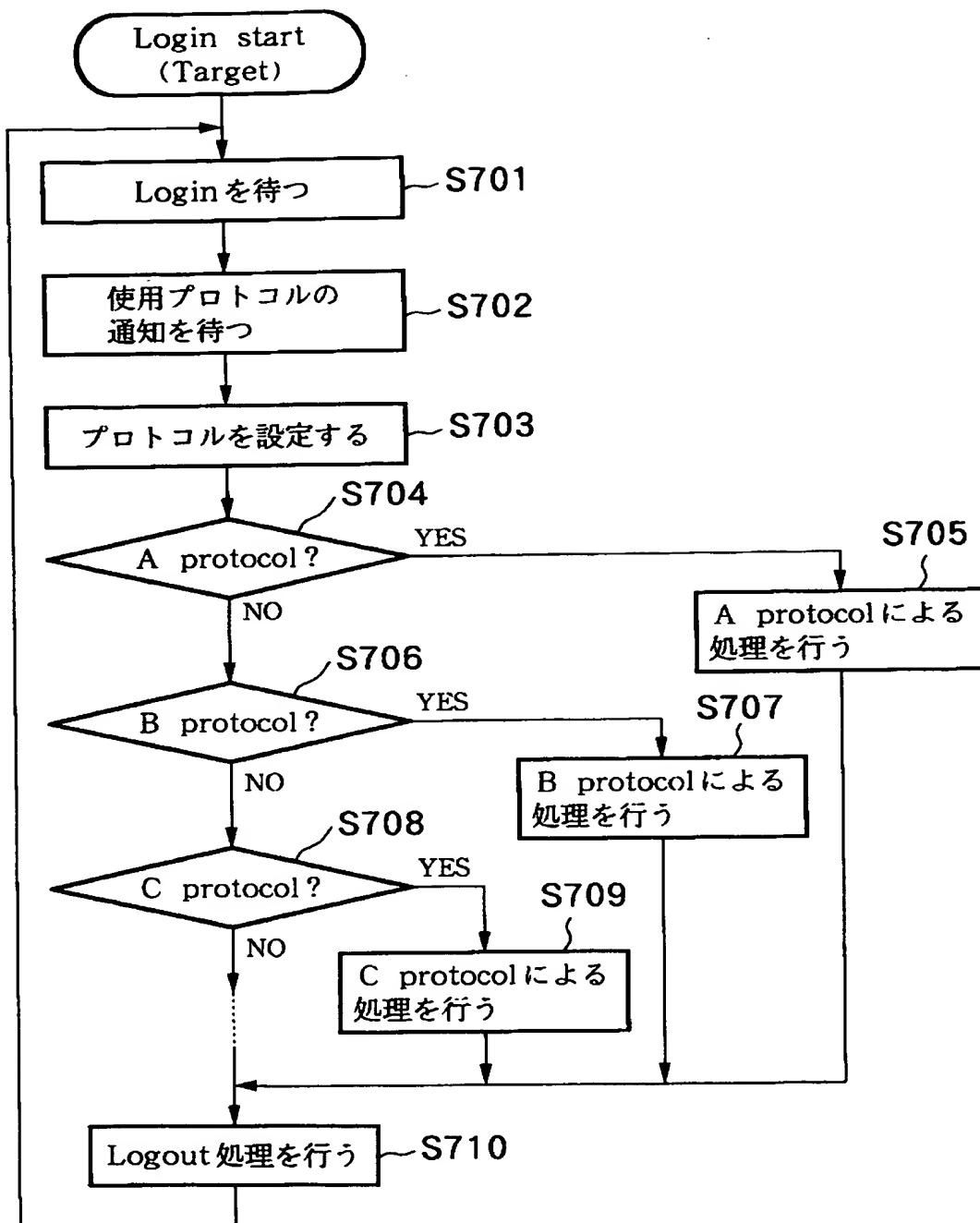
【図37】



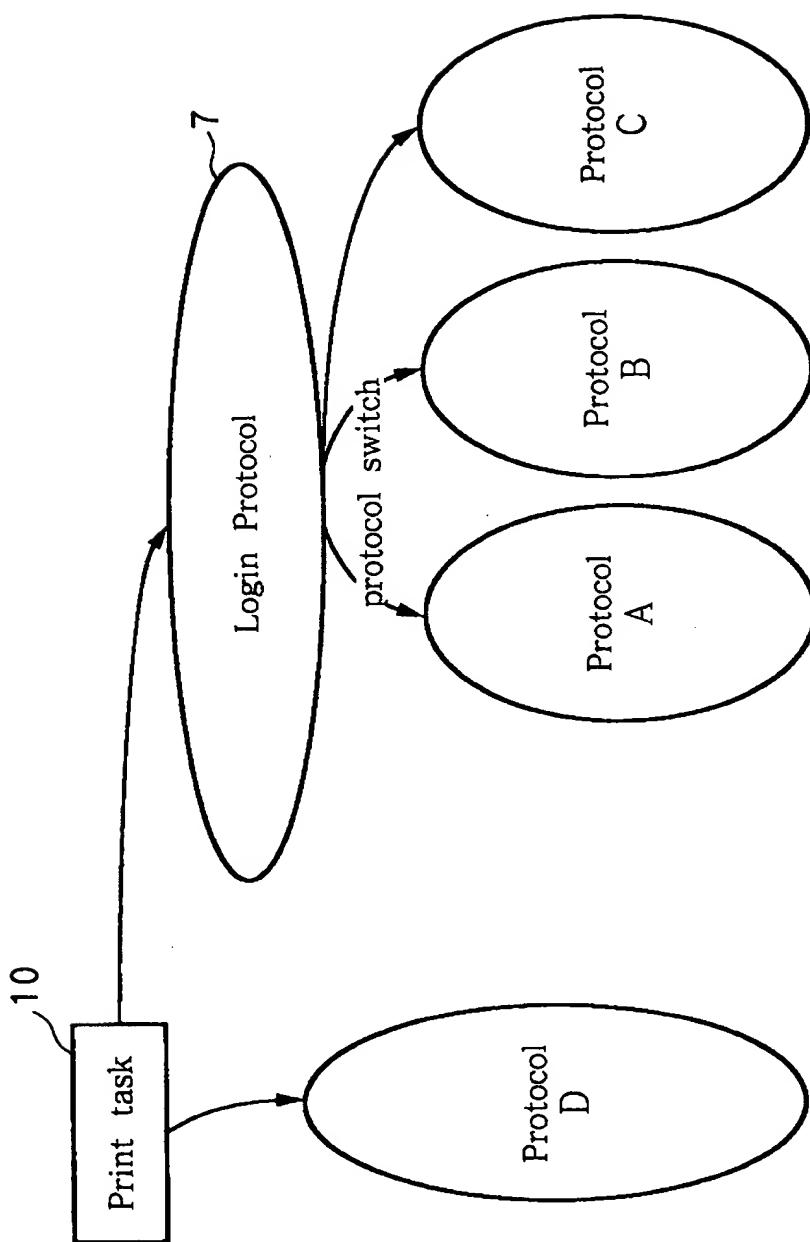
【図38】



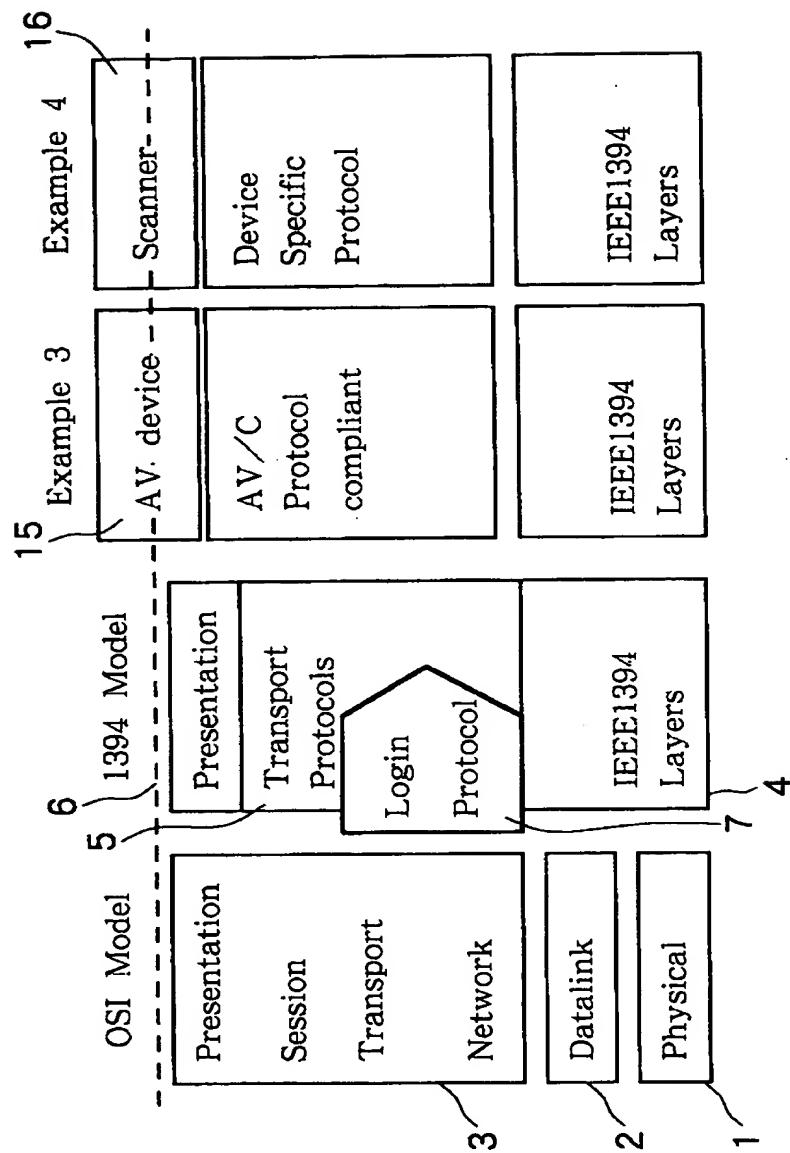
【図39】



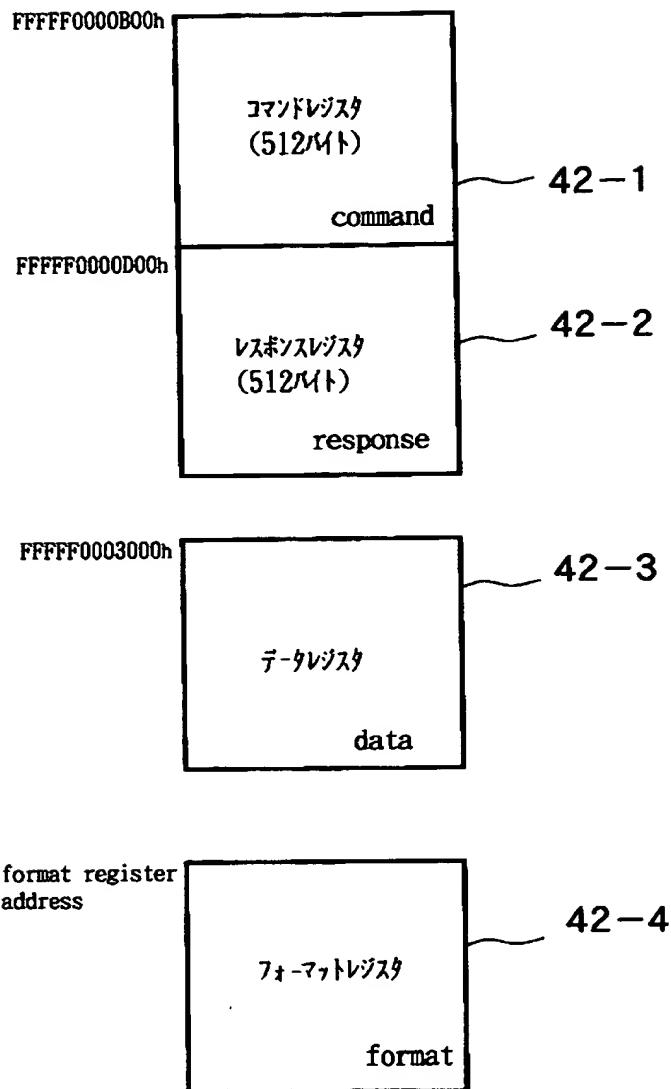
【図40】



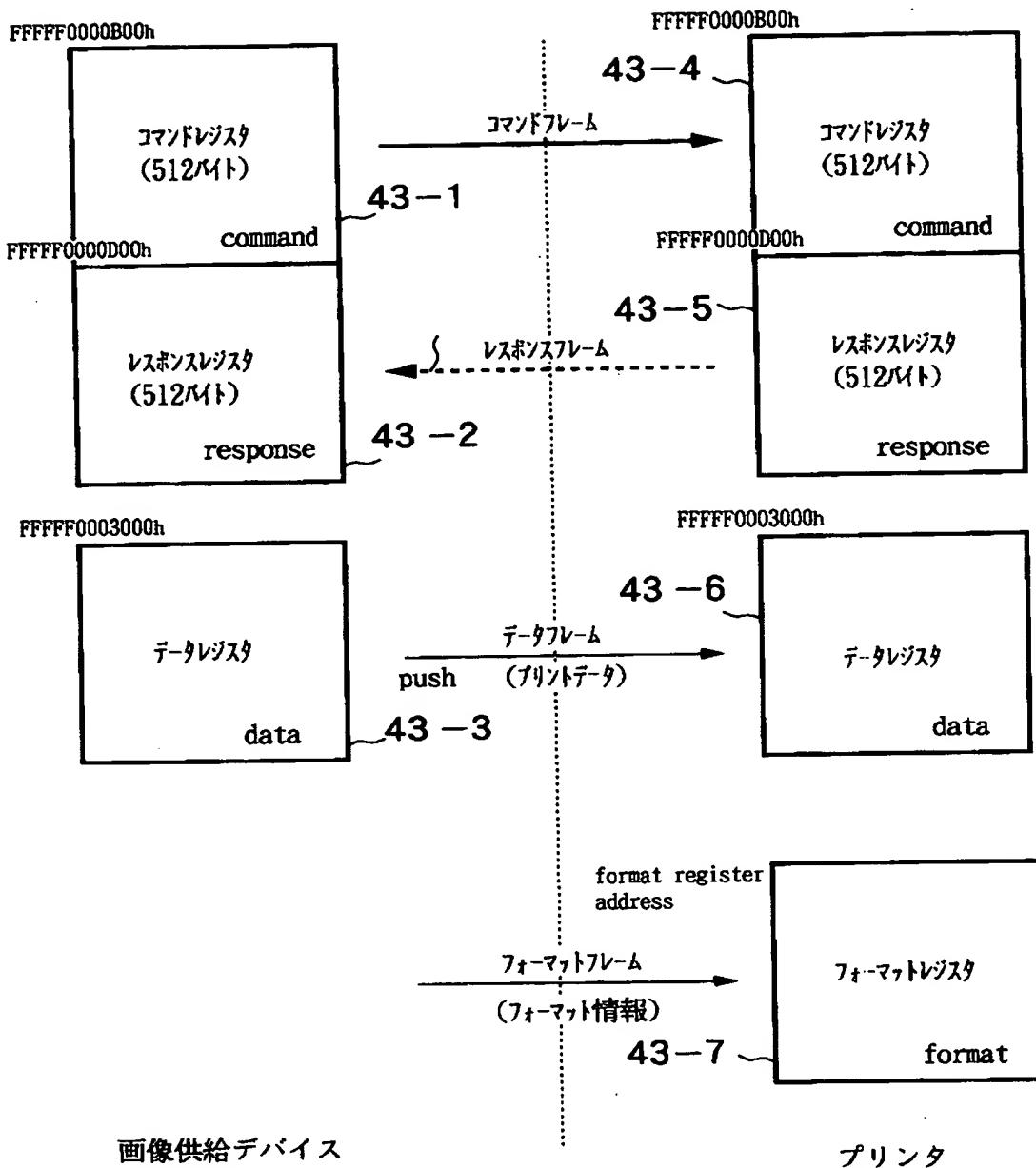
【図4.1】



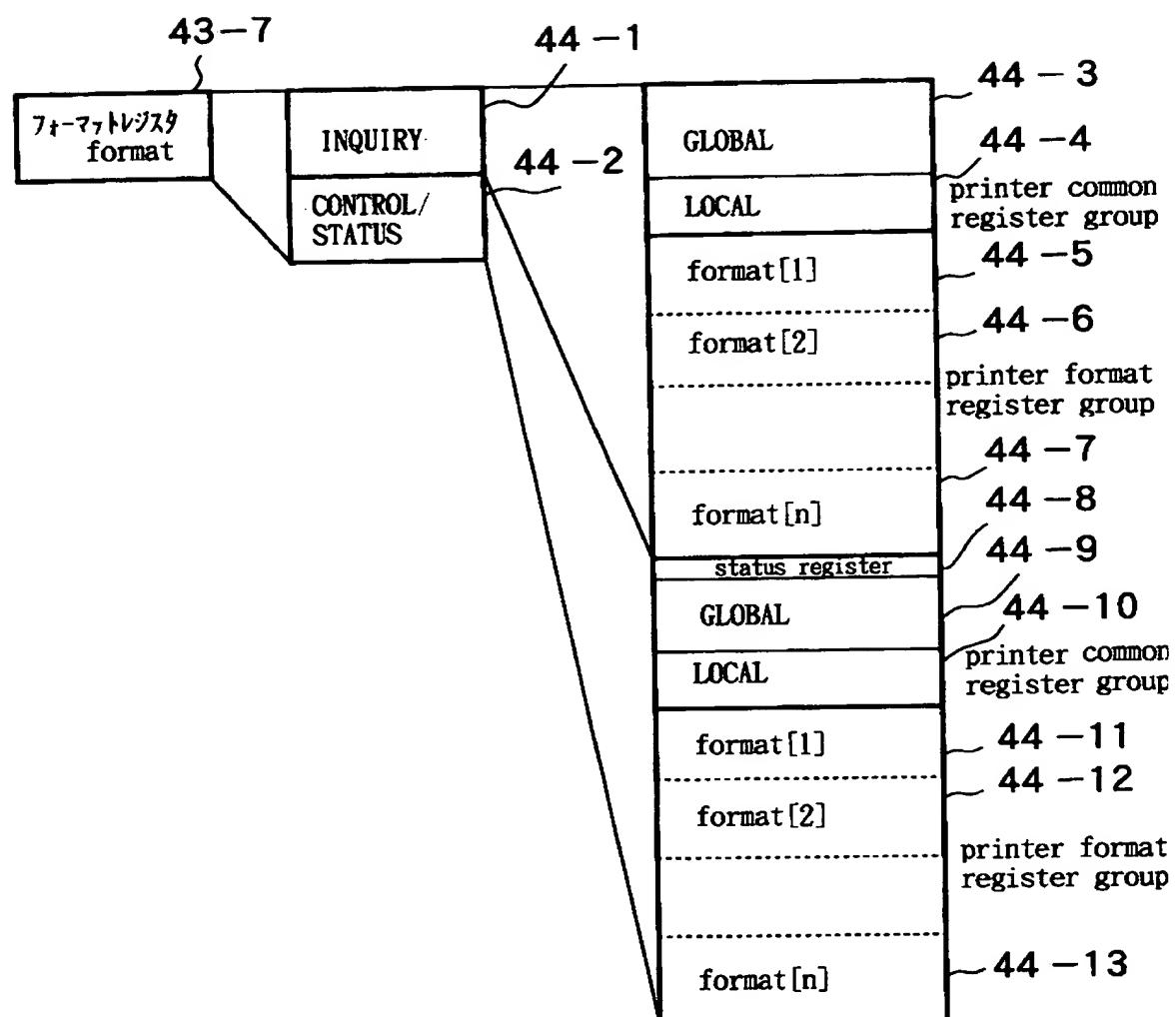
【図42】



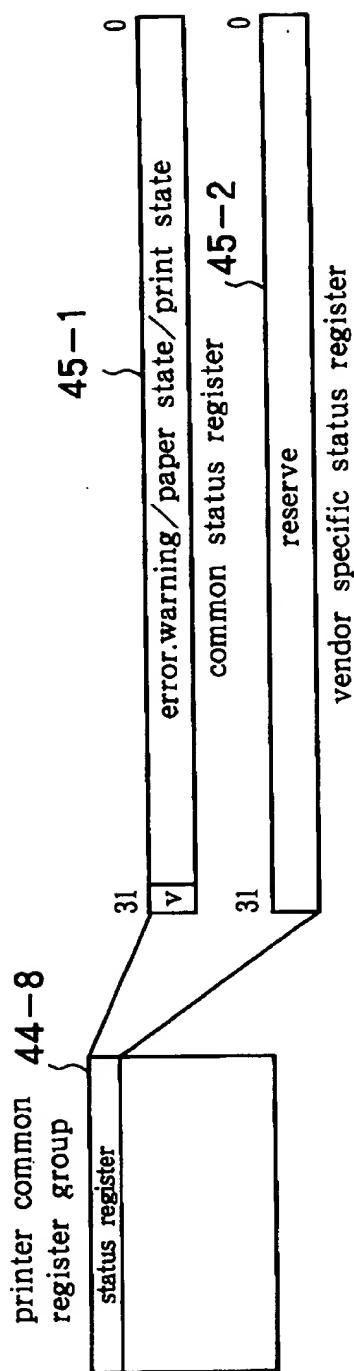
【図43】



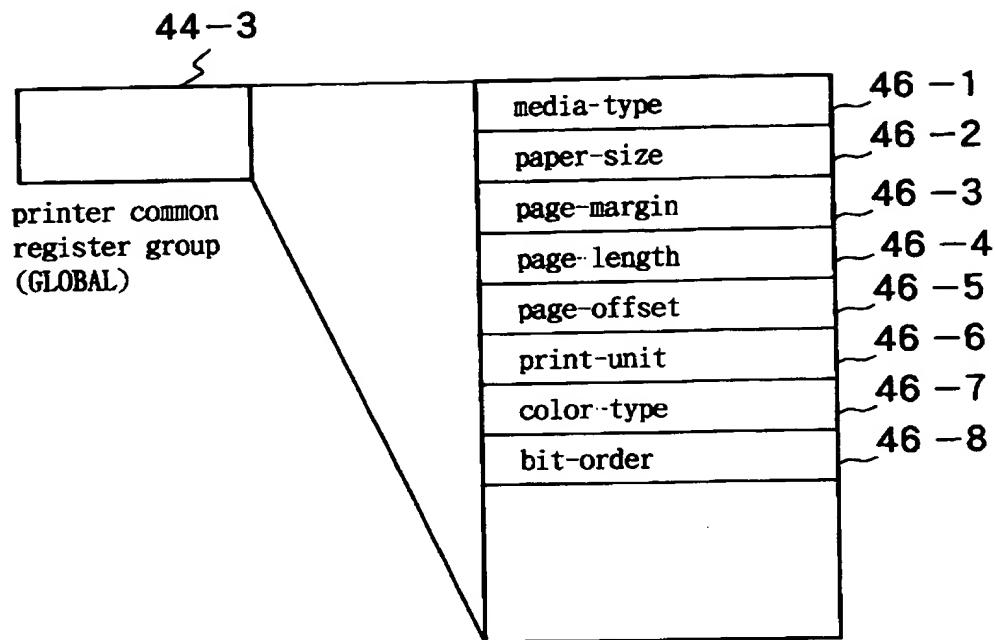
【図44】



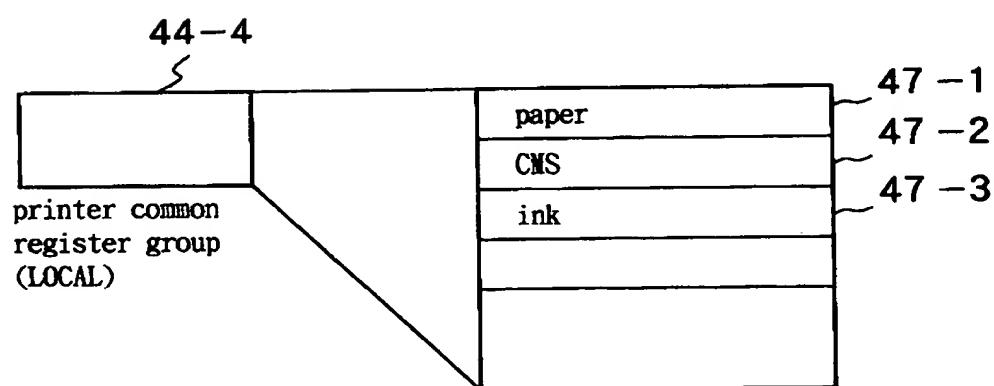
【図45】



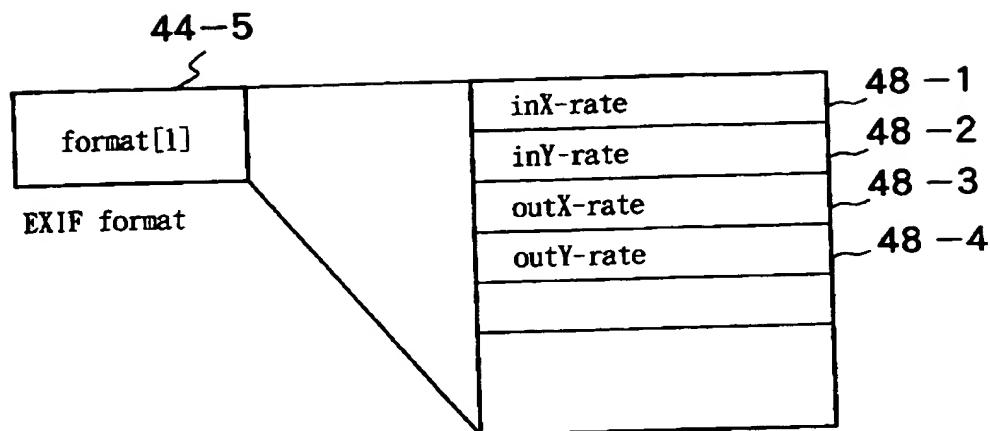
【図4 6】



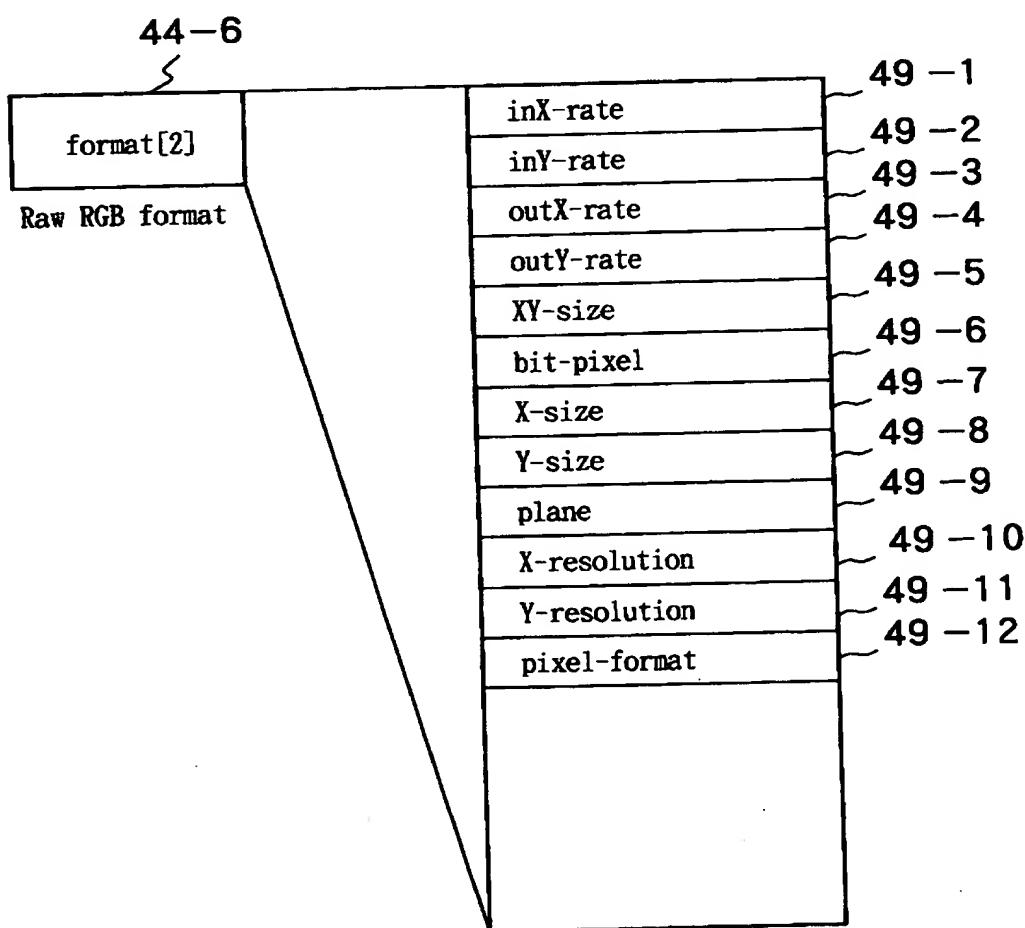
【図4 7】



【図48】



【図49】



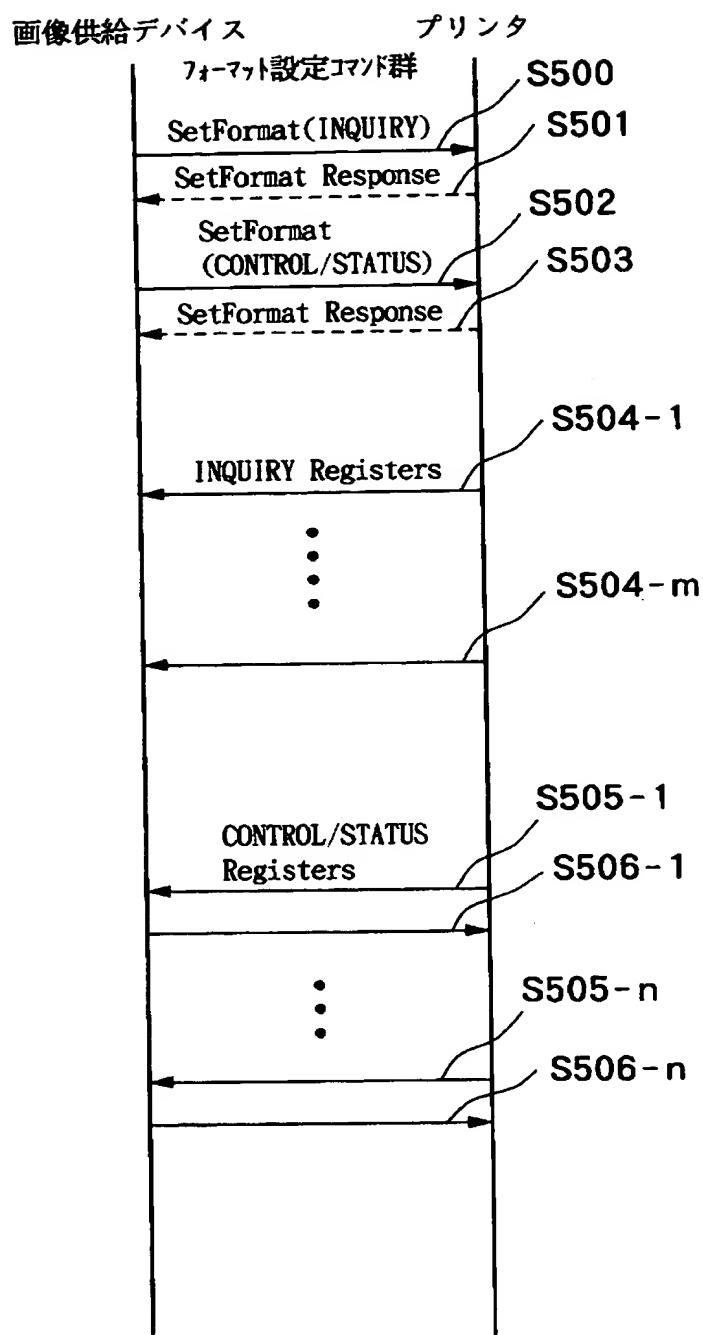
【図50】

コマンド分類	command	response	
ステータス	Getstatus	GetstatusResponse	~ 50-1
コントロール	PrintReset	PrintResetResponse	~ 50-2
	PrintStart	PrintStartResponse	~ 50-3
	PrintStop	PrintStopResponse	~ 50-4
	InsertPaper	InsertPaperResponse	~ 50-5
	EjectPaper	EjectPaperResponse	~ 50-6
	CopyStart	CopyStartResponse	~ 50-7
	CopyEnd	CopyEndResponse	~ 50-8
ブロック /バッファ	BlockSize	BlockSizeResponse	~ 50-9
	BlockAddress	BlockAddressResponse	~ 50-10
	FreeBlock	FreeBlockResponse	~ 50-11
	WriteBlocks	WriteBlocksResponse	~ 50-12
	BufferConfig	BufferConfigResponse	~ 50-13
	SetBuffer	SetBufferResponse	~ 50-14
チャネル	OpenChannel	OpenChannelResponse	~ 50-15
	CloseChannel	CloseChannelResponse	~ 50-16
転送	TransferMethod	TransferMethodResponse	~ 50-17
フォーマット	SetFormat	SetFormatResponse	~ 50-18
ログイン	Login	LoginResponse	~ 50-19
	Logout	LogoutResponse	~ 50-20
	Reconnect	ReconnectResponse	~ 50-21
データ	WriteBlock		~ 50-22
	WriteBuffer		~ 50-23
	PullBufer		~ 50-24

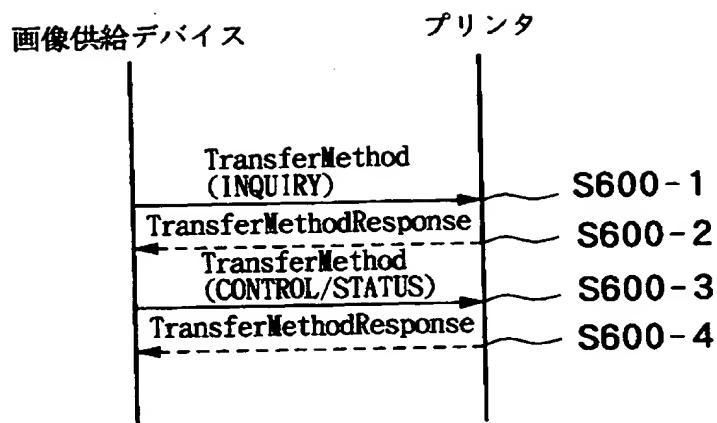
【図51】

EXIF(TIFF, JPEG)	EXIFの非圧縮データと圧縮データ
TIFF/EP	TIFFの拡張バージョン
REG	REGロードイメージ
YUV	YUVロードイメージ
YCrCb	YCrCbロードイメージ
CMYK	CMYKロードイメージ
Vender Specific	ベンダ定義

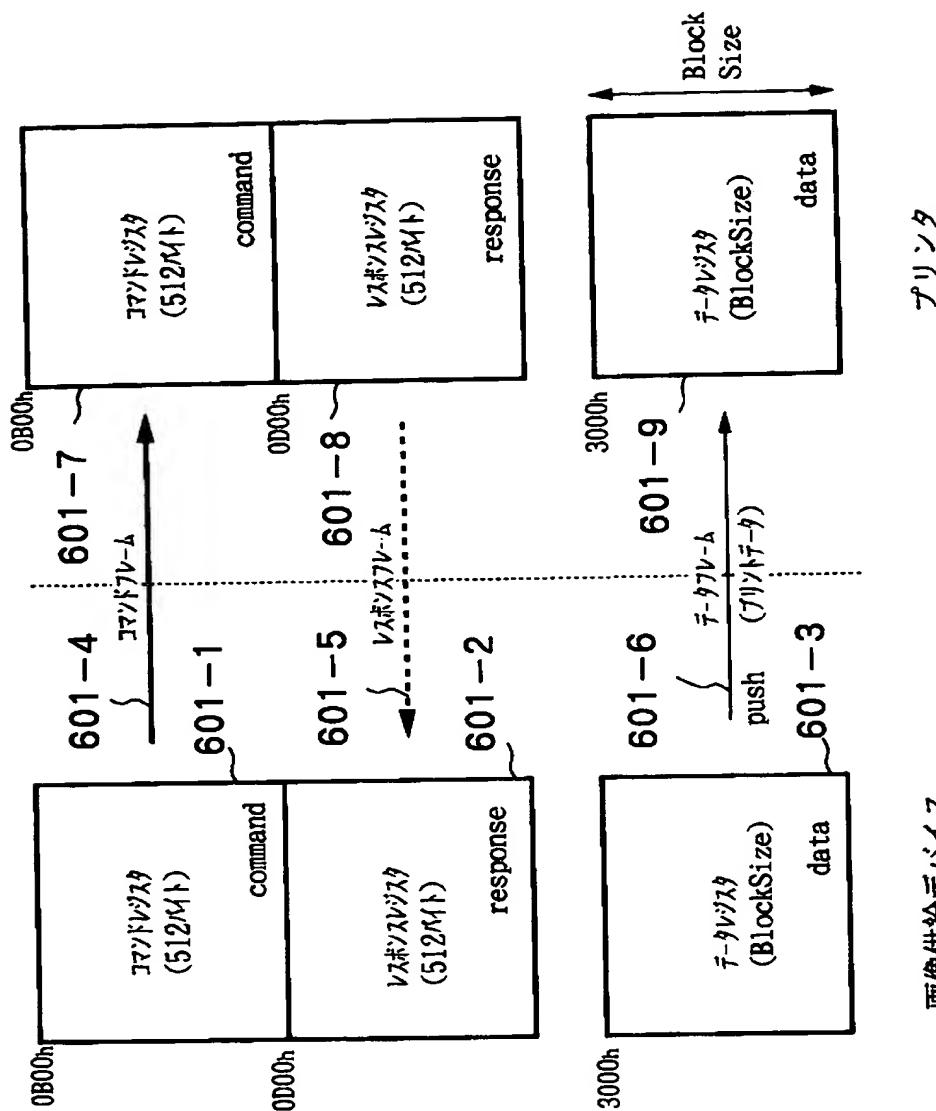
【図52】



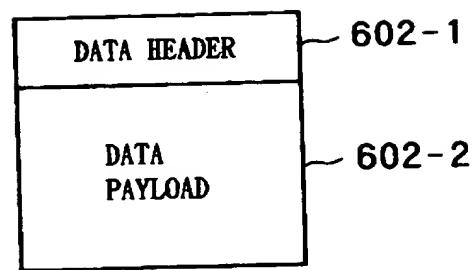
【図53】



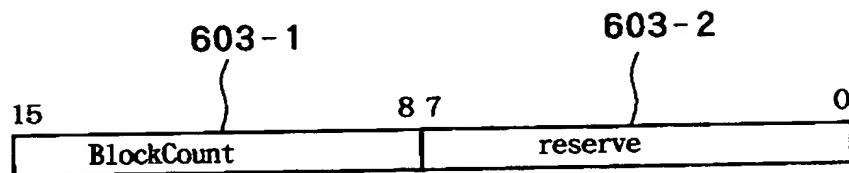
【図54】



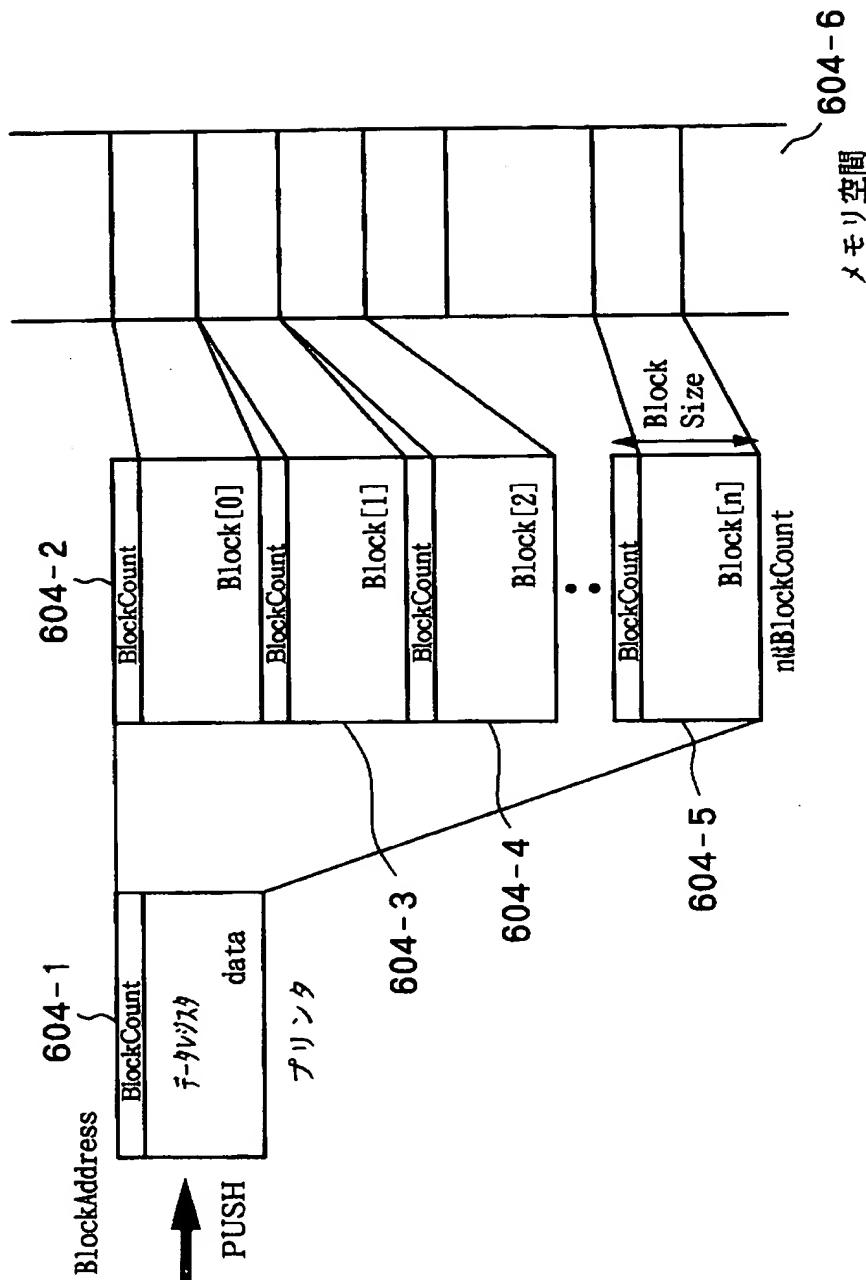
【図55】



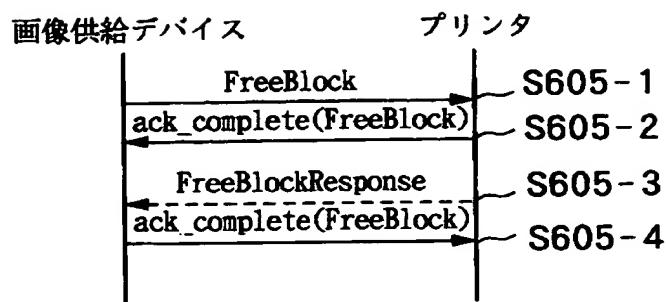
【図56】



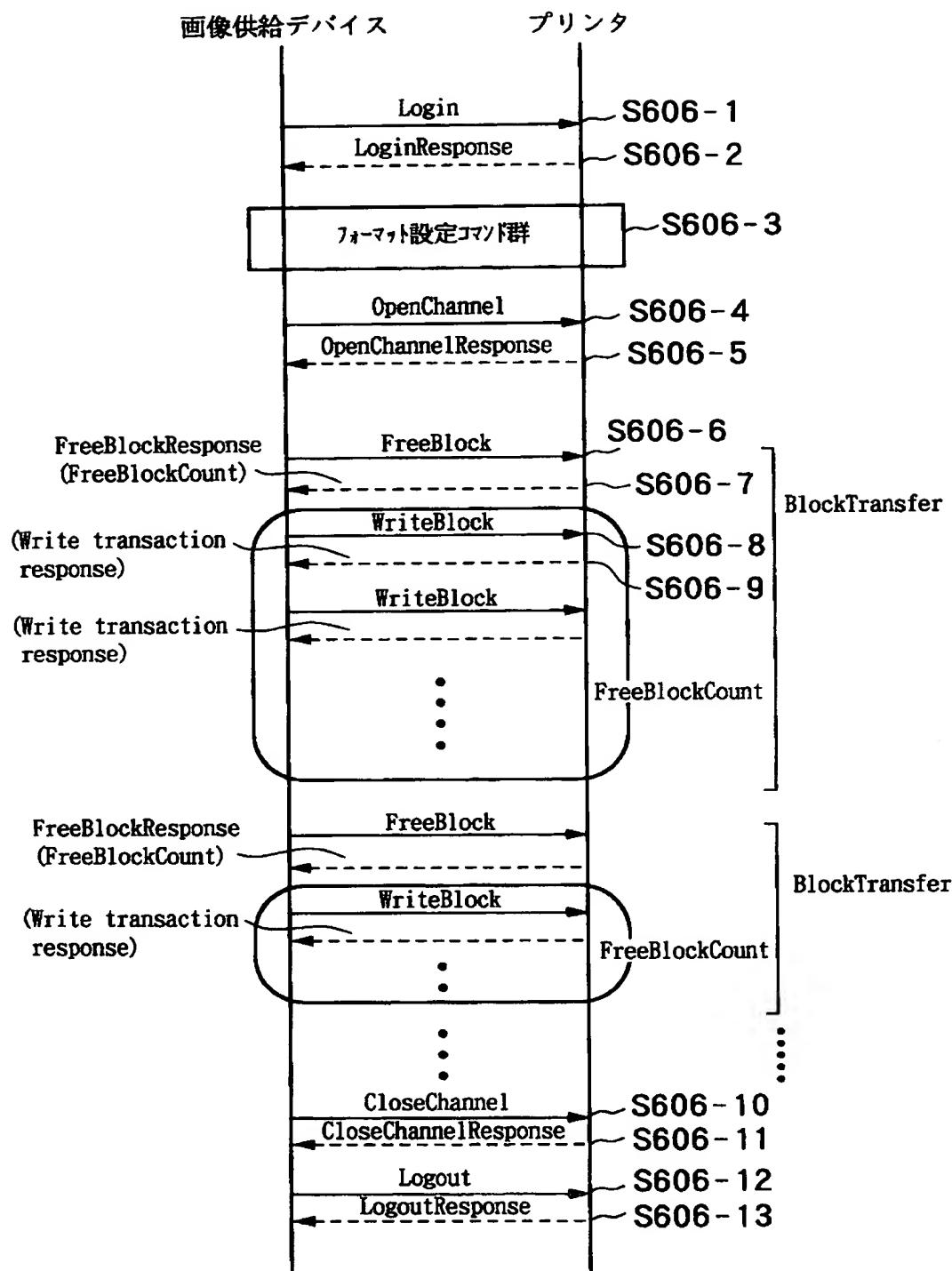
【図57】



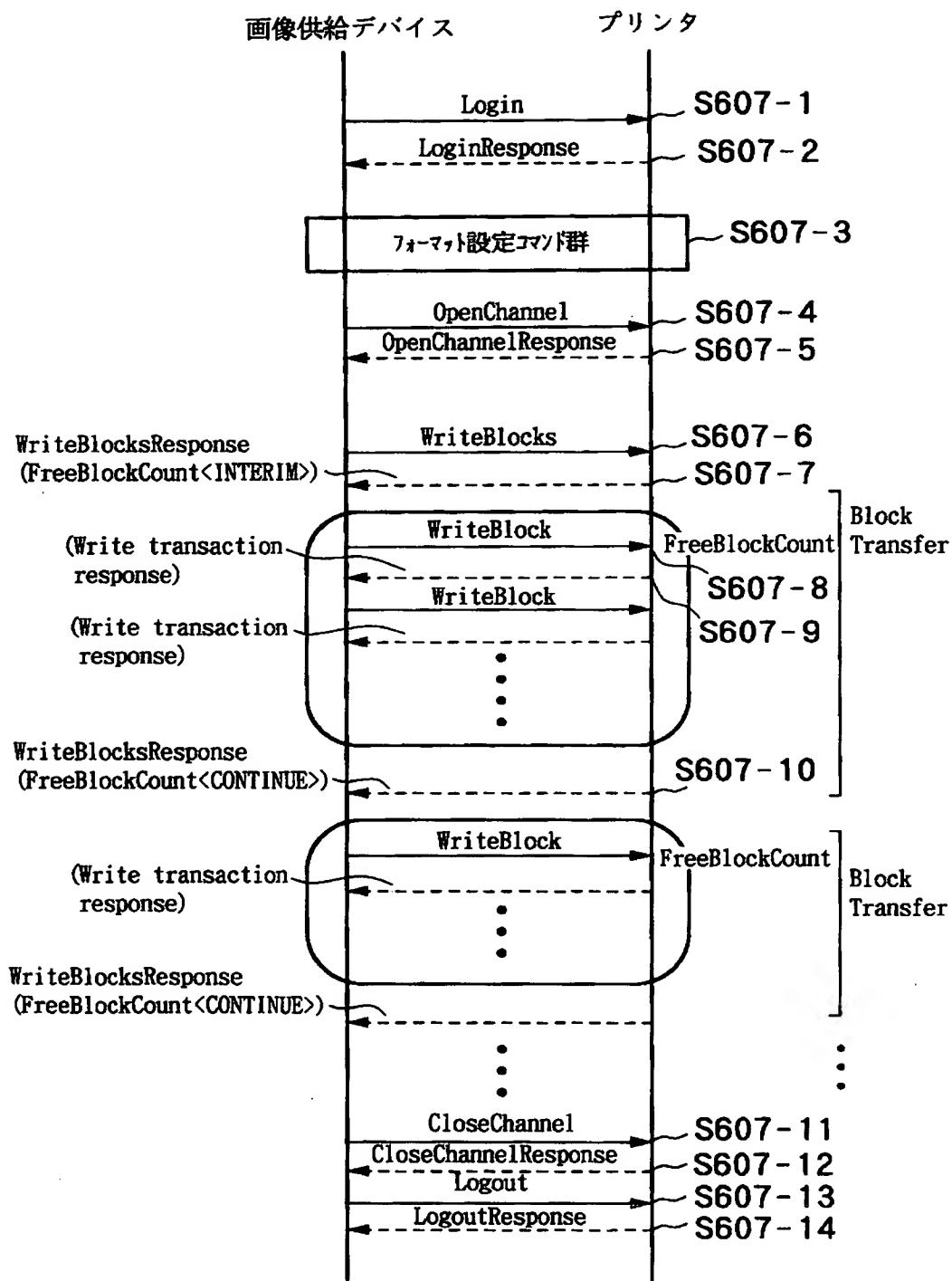
【図58】



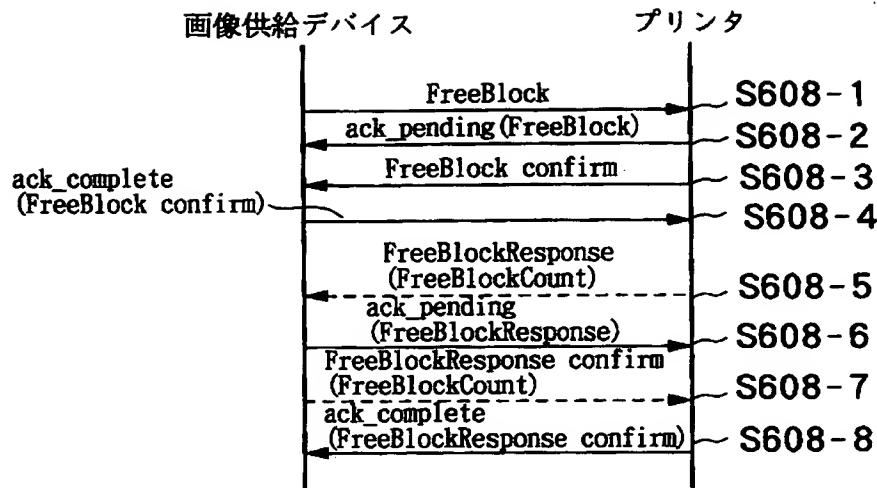
【図59】



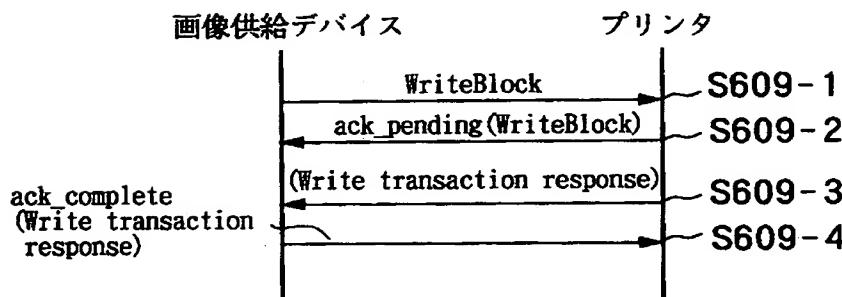
【図60】



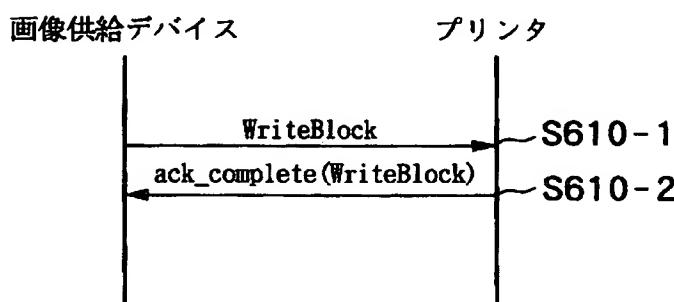
【図 6 1】



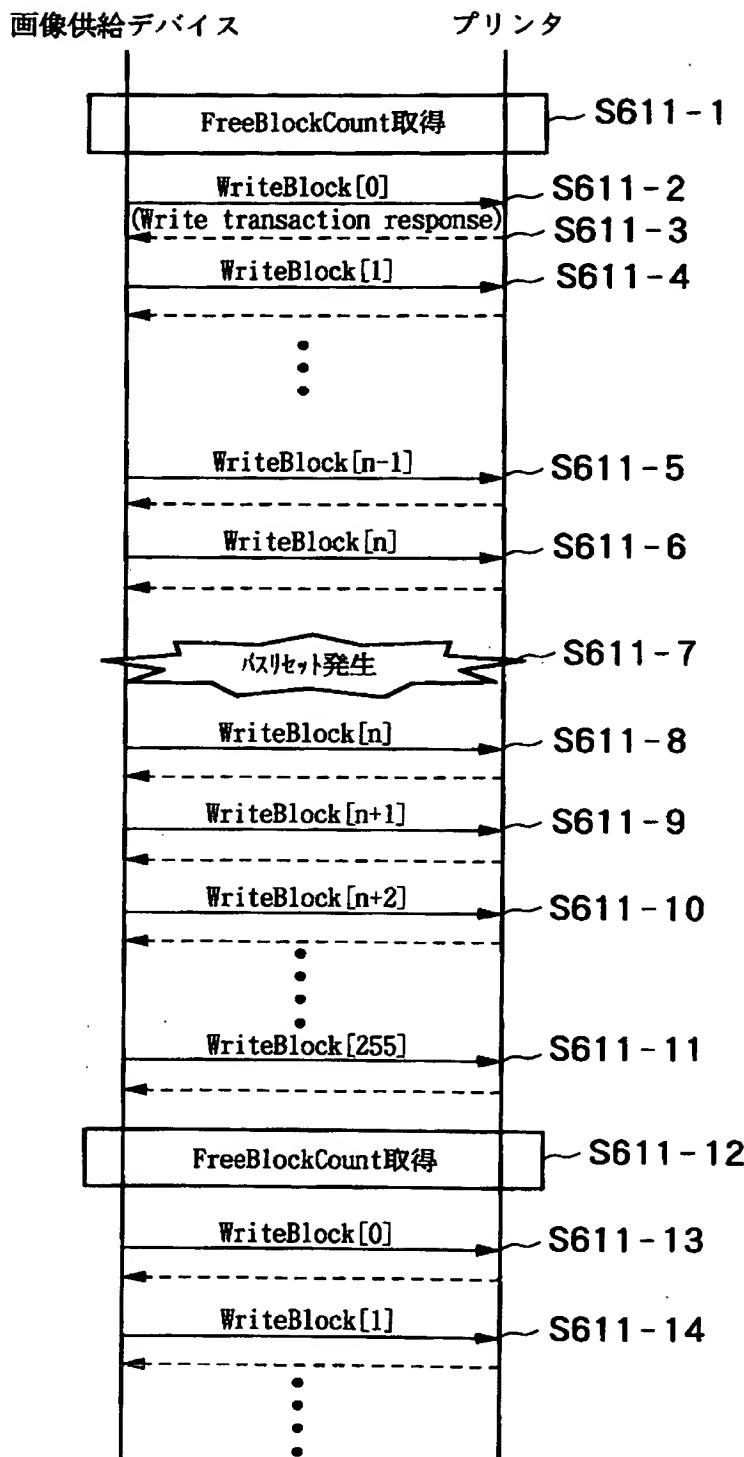
【図 6 2】



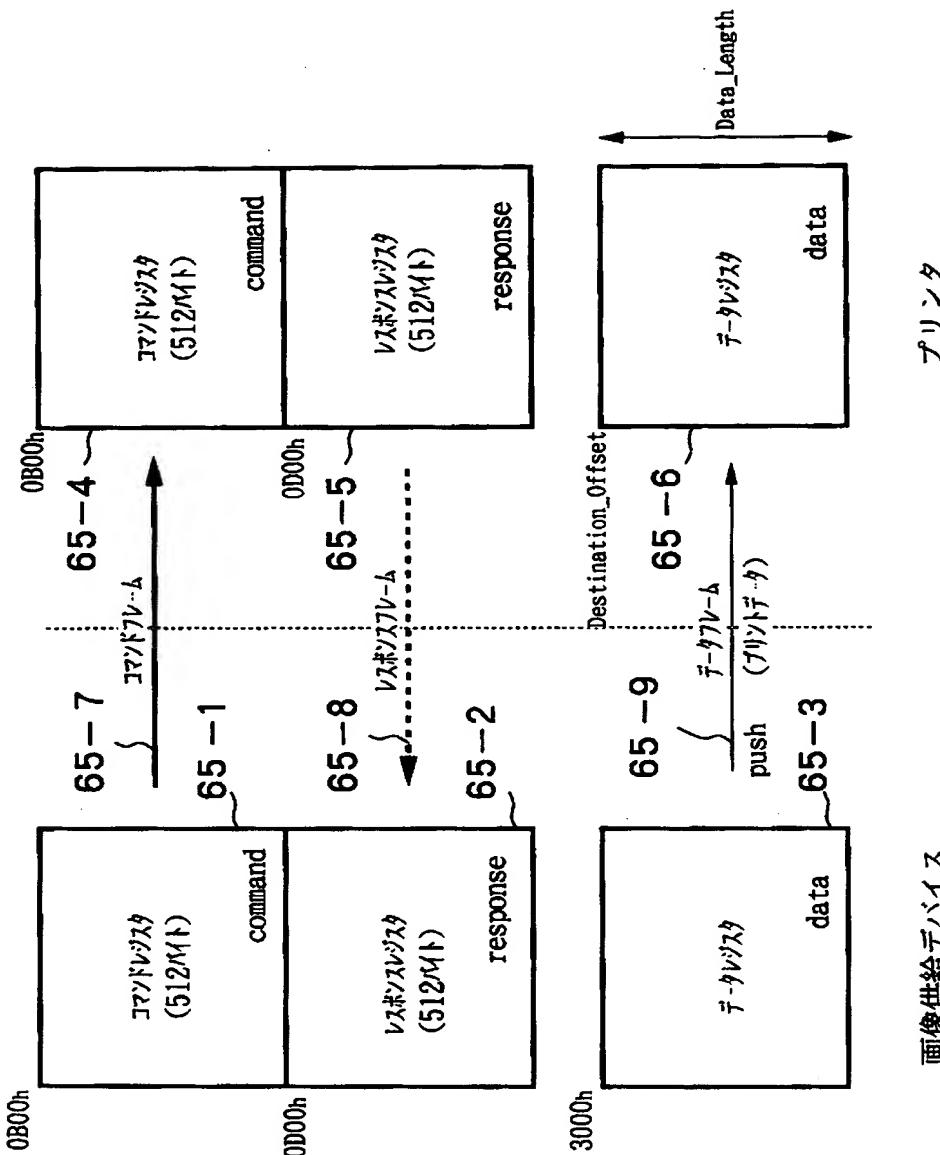
【図 6 3】



【図64】



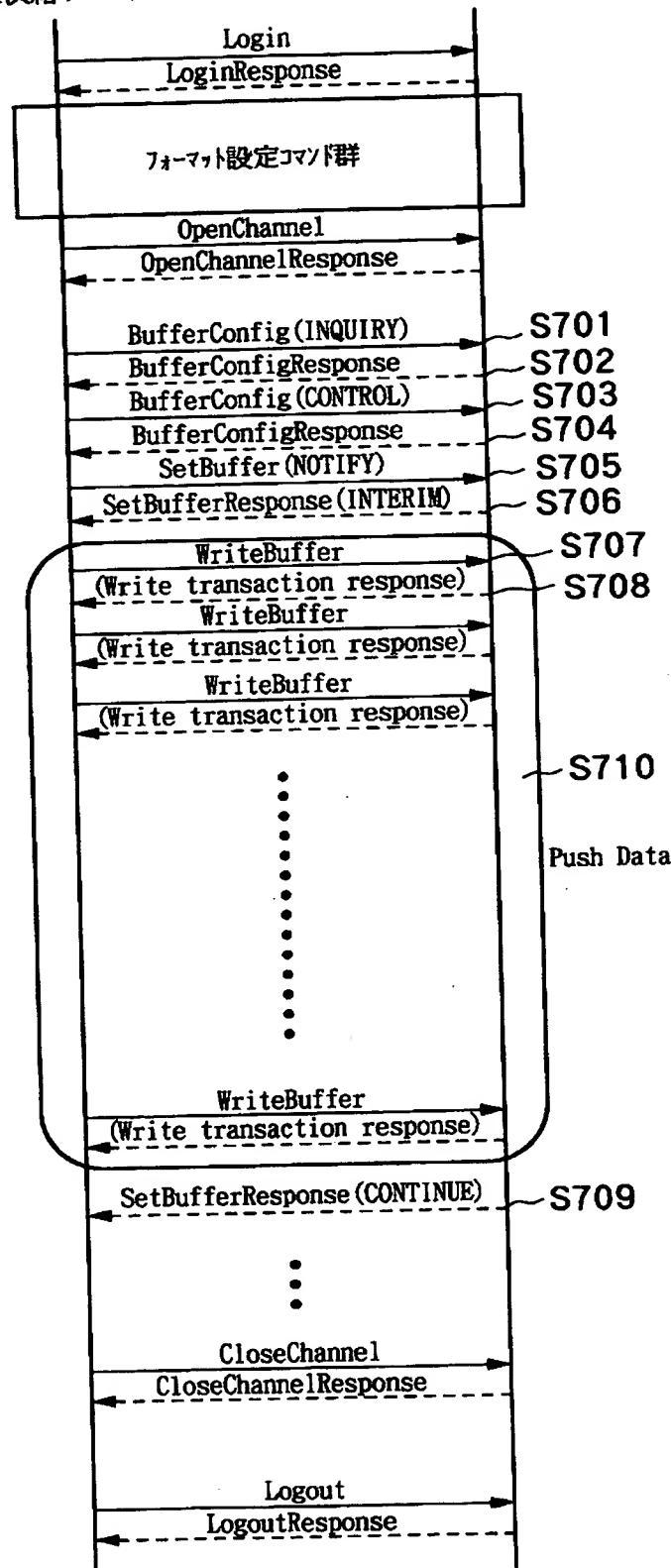
【図 6 5】



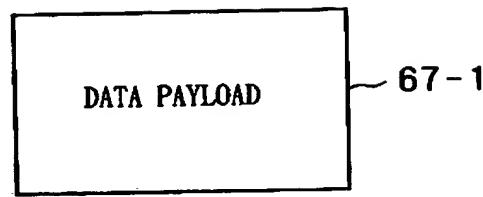
【図 6 6】

画像供給デバイス

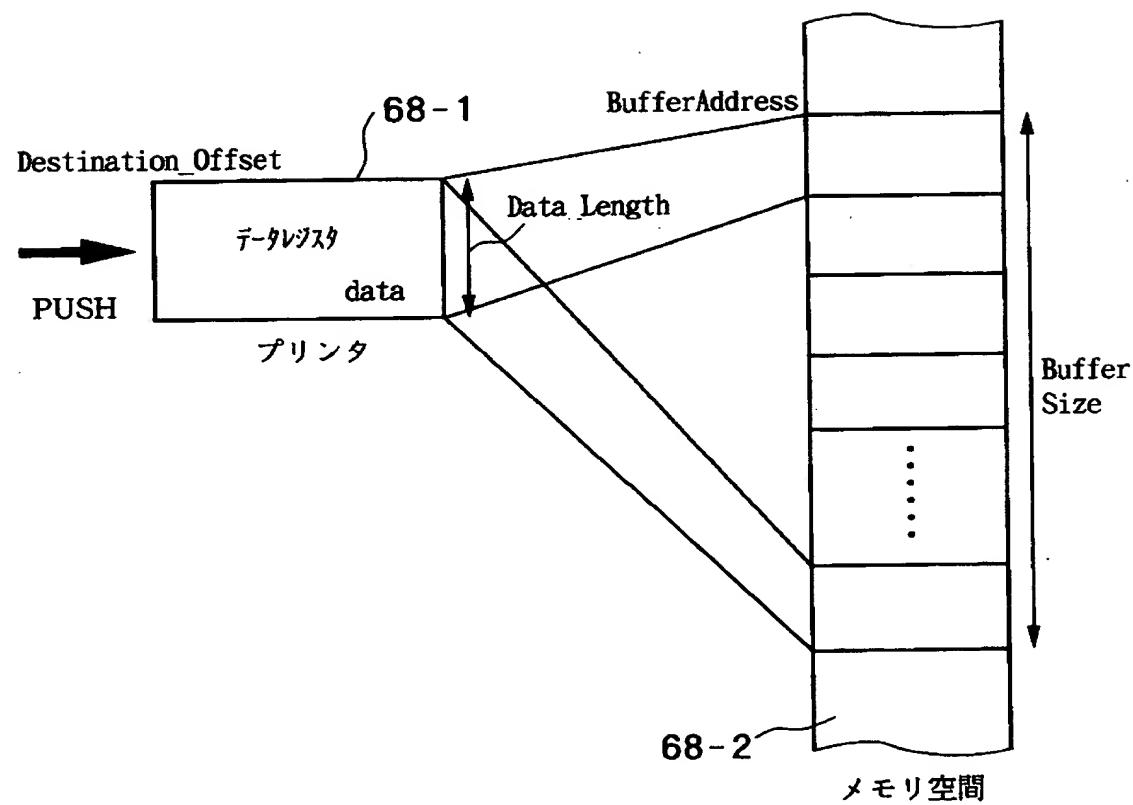
プリンタ



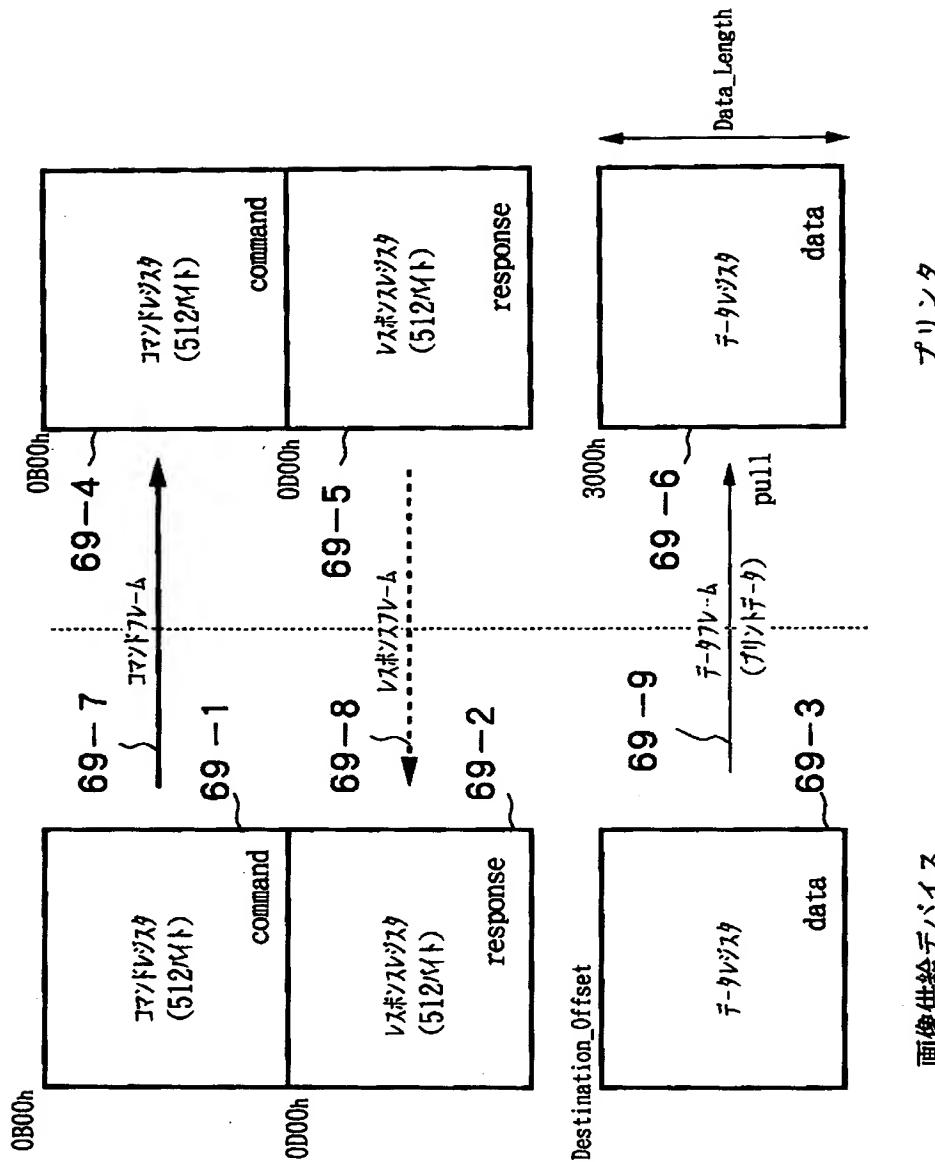
【図 6 7】



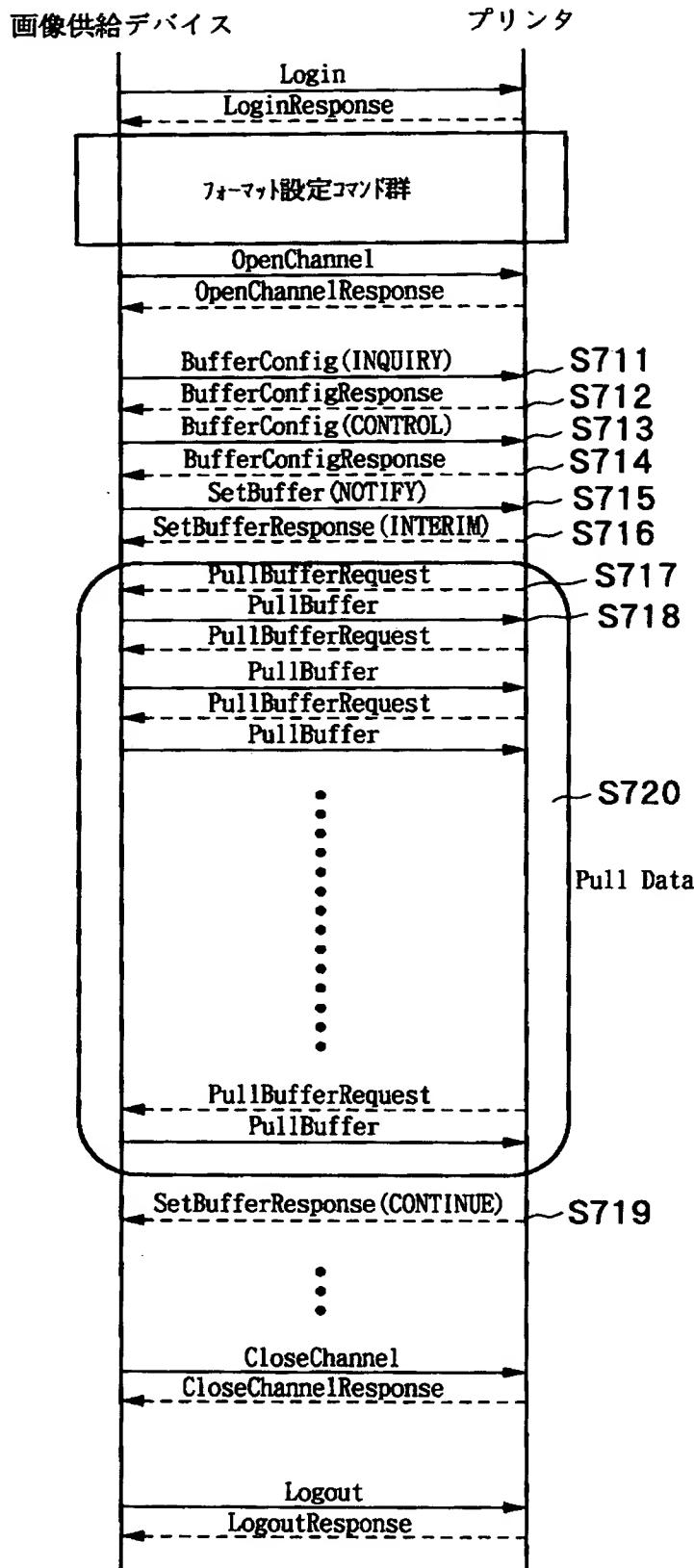
【図 6 8】



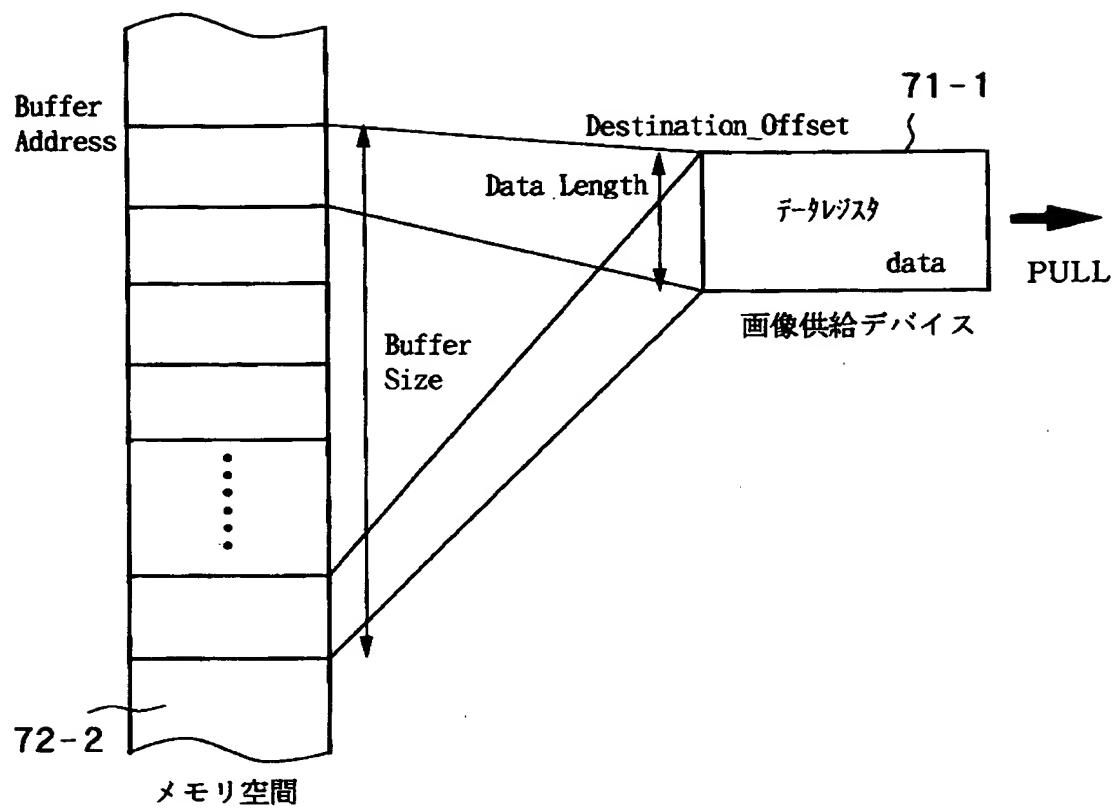
【図 6 9】



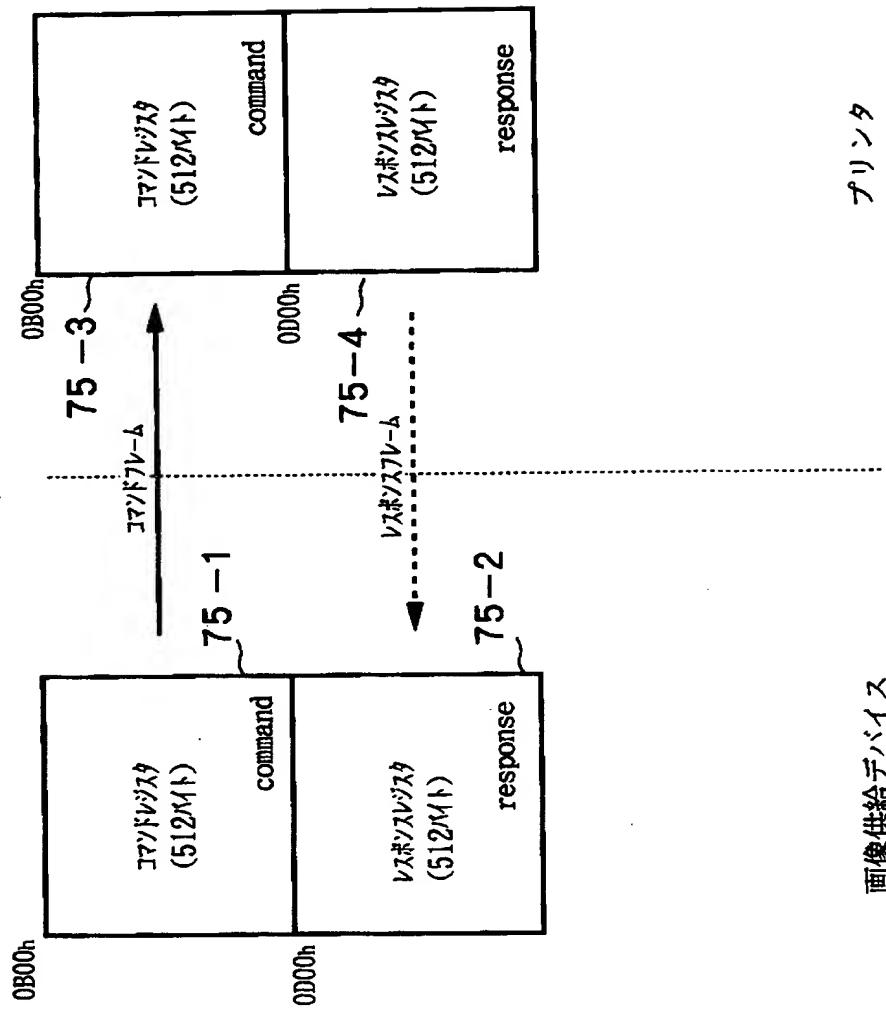
【図 7 0】



【図 7 1】

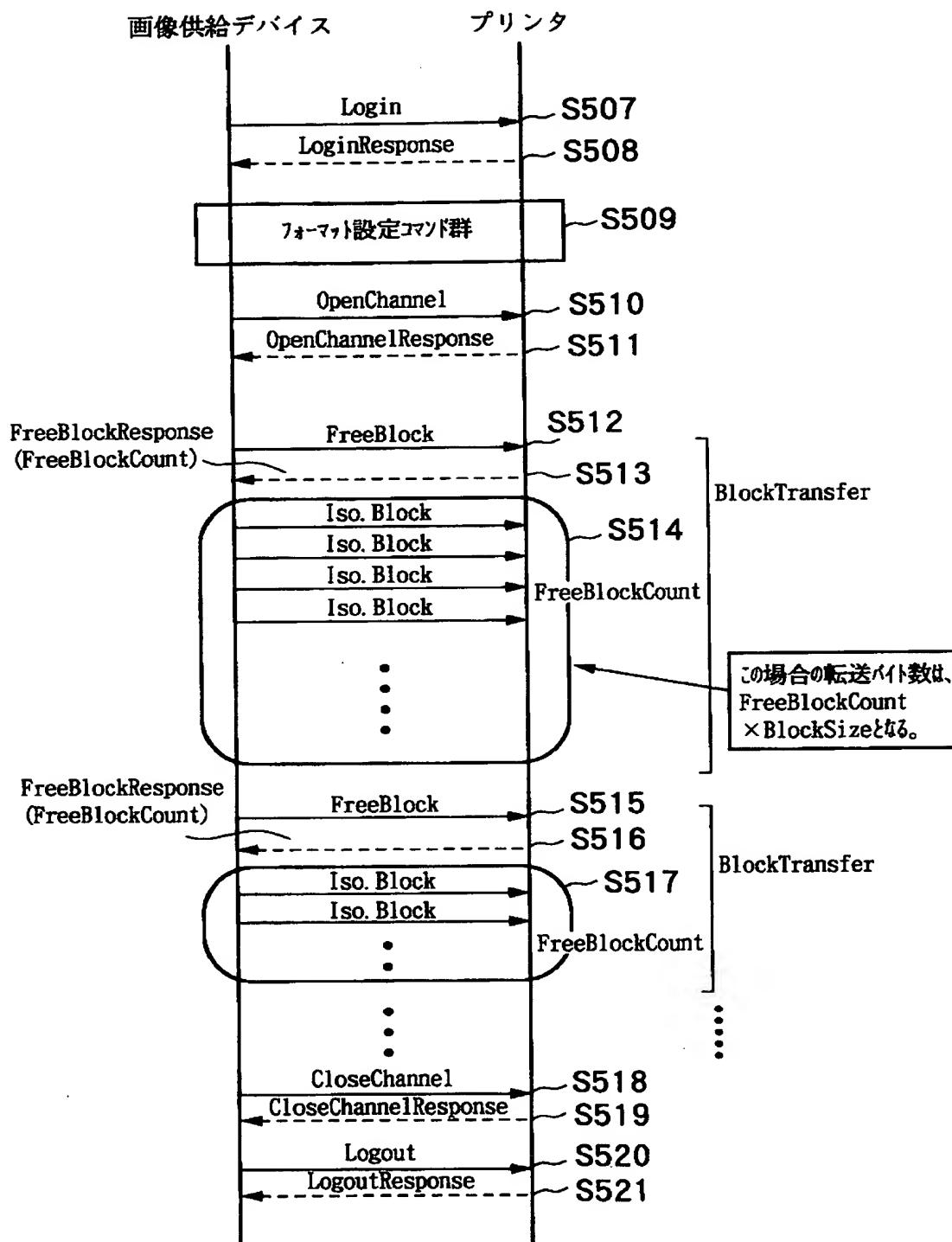


【図72】



プリント
画像供給デバイス

【図73】



【書類名】 要約書

【要約】

【課題】 画像供給デバイスから出力される画像データは、ホストコンピュータにより印刷データに変換されてプリンタにより印刷されるものであるから、1394シリアルバスなどにより画像供給デバイスとプリンタを直結したとしても、ホストコンピュータが無ければ印刷を行うことができない。

【解決手段】 1394シリアルバスにより画像供給デバイス101とプリンタ102を直結し、画像供給デバイス101からプリンタ102へコマンドを送らせ、プリンタ102から画像供給デバイス101へコマンドに対するレスポンスを返させ、そのレスポンスに含まれる情報に基づき、画像供給デバイス101からプリンタ102へ画像データを送らせる。

【選択図】 図1